

Grundbegriffe der Theoretischen Informatik

Folien zur Vorlesung von E. Best

Oktober 2002 – Februar 2003

Wer sind wir?

- Prof. Eike Best, Abteilung Parallele Systeme
A3-2-207, Tel.: 798-2973
eike.best@informatik.uni-oldenburg.de
Sprechstunde: Mo 14-16 oder nach Vereinbarung.
- Dr. Harro Wimmel, Abteilung Parallele Systeme
A3-2-215, Tel.: 798-3495
harro.wimmel@informatik.uni-oldenburg.de
Sprechstunde: Mo 14-16 oder nach Vereinbarung.

Wer sind Sie?

- Studierende der Informatik im 3. Semester (oder ...),
- ... die den Schein für das Modul Theoretische Informatik 2 er-gattern wollen.

Was müssen Sie tun?

- An der Vorlesung teilnehmen.
- An den Übungen teilnehmen.
- Die Übungsaufgaben bearbeiten (max. 3er Gruppen).
- 40% der Aufgaben richtig bearbeitet haben.
- Das Modul belegen (gegen Semesterende).
Dies ist nur möglich, wenn genügend Aufgaben gelöst wurden.
- Die Klausur bestehen (am Montag, 10.2.2003, 9:00 s.t., A14 HS 1+2, bzw. am Freitag, 11.4.2003, 9:00 s.t., A14 HS 1+2).

Vorlesung

- Dienstags, 10-12, A7 – HS G
- Donnerstags, 10-11, A14 – HS 2
- **In der ersten Woche:** Donnerstags 10-12, A14 – HS 2!
- Je nach Bedarf werde ich am Donnerstagstermin auch 1-2 Dinge erzählen, die als Ergänzung dienen, z.B. zusätzliche Beispiele geben.

Übungen

- Übungsblätter werden jeden Dienstag bei der Vorlesung ausgeteilt (beginnend heute).
- **Abgabetermin:** bis darauffolgenden Montag, 12:00, roter Bauteil, 2. Stockwerk, Postfach.

Übungsgruppen

Dienstag	8-9	A4 4-441	Henning Dierks
Dienstag	9-10	A4 4-441	Hans Fleischhack
Mittwoch	8-9	A4 5-516	Stephanie Kemper
Mittwoch	9-10	A4 5-516	Stephanie Kemper
Mittwoch	12-13	A1 0-007	Harro Wimmel
Mittwoch	13-14	A1 0-007	Harro Wimmel
Freitag	12-13	A4 5-516	Jochen Hoenicke
Freitag	13-14	A4 2-221	Jochen Hoenicke.

Einteilung in Übungsgruppen

- Einteilungszettel werden ausgeteilt (nur heute!).
- Eintragen:
 1. Wer zusammen in 1 Gruppe die Aufgaben bearbeitet.
 2. Prioritätenliste für Übungstermine.

!!BITTE DRUCKSCHRIFT!!
- Einteilung nach Einsammeln, Zufallsgenerator falls nötig.
- Aushang bis spätestens Donnerstag, 17.10.2002, 10 Uhr.

Unterlagen?

- GTI-Skript: siehe
parsys.informatik.uni-oldenburg.de/best/lecture-notes.html
parsys.informatik.uni-oldenburg.de/teaching/gti_ws0203/data.html
- Uwe Schöning: „Theoretische Informatik – kurz gefasst“. EUR 17,95.
- John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman: „Introduction to Automata Theory, Languages, and Computation, 2/E“. USD 89,00.
- Übungszettel (auch im Netz).
- Folien (*vielleicht* auch im Netz, aber erst später).

Kommen wir zur Sache!

Was wird in GTI gelehrt?

- Teil 1: Automatentheorie und formale Sprachen
- Teil 2: Berechenbarkeitstheorie
- Teil 3: Komplexitätstheorie

Automatentheorie und formale Sprachen

- ... beschäftigen sich mit der Frage, wie man **formale Sprachen** (nicht natürliche Sprachen) und **berechenbare Funktionen** klassifizieren und erkennen kann.
- Die Klassifizierung geschieht durch unterschiedliche Grammatikklassen und Computermodelle („Automaten“).
- Für die wichtigsten Klassen der Sprachenhierarchie hat man jeweils eine Grammatikklasse und ein Automatenmodell.

Buchstaben

- Sprachen sind Wortmengen.
- Wörter bestehen aus Buchstaben.
- Buchstaben (Zeichen, Symbole) sind z.B.
 $a, b, c, \dots, A, B, C, \dots,$
 $0, 1, \dots, \text{ä}, \text{ö}, \dots, \text{ℵ}, \$, \%, \dots,$
 $(a, b), [1, z], (a, [x, y]), \dots$
- Es gibt eine **abzählbar unendliche** Menge von Symbolen als **Grundvorrat** für alle Symbole, die wir benötigen.

Abzählbarkeit

- Alle endlichen Mengen (insbesondere \emptyset) sind abzählbar.
- Die natürlichen Zahlen \mathbb{N} sind abzählbar unendlich.
- Das Kreuzprodukt $\mathbb{N} \times \mathbb{N}$ (die Menge aller Paare natürlicher Zahlen) ist abzählbar unendlich. *Beweis?*
- Die Menge $\mathbb{N} \rightarrow \mathbb{N}$ aller Funktionen von \mathbb{N} nach \mathbb{N} ist **nicht** abzählbar. *Beweis?*

Beweis der Abzählbarkeit von $\mathbb{N} \times \mathbb{N}$

$\mathbb{N} \times \mathbb{N}$	0	1	2	...
0	(0, 0)	(0, 1)	(0, 2)	...
1	(1, 0)	(1, 1)	(1, 2)	...
2	(2, 0)	(2, 1)	(2, 2)	...
\vdots	\vdots	\vdots	\vdots	

Bringe die Elemente der unteren rechten Viertelebene in eine Reihenfolge $0, 1, 2, \dots$

Nicht: $(0, 0), (0, 1), (0, 2), \dots, (1, 0), (1, 1), (1, 2), \dots$

Aber: $(0, 0), (0, 1), (1, 0), (0, 2), (1, 1), (2, 0), (0, 3), (1, 2), \dots!$

D.h.: bildlich gesprochen, eine **Diagonalisierung** „von rechts oben nach links unten“, oder, arithmetisch gesprochen, nach Summe der beiden Einzelzahlen (und „lexikalisch“ bei gleicher Summe).

Diese Abzählung vermittelt eine **Bijektion** zwischen \mathbb{N} und $\mathbb{N} \times \mathbb{N}$.

Effektivität der Diagonalabzählung

Diese Bijektion ist auch **effektiv**, d.h., es gibt zwei Algorithmen:

- Einer, der zu einem Paar natürlicher Zahlen diejenige natürliche Zahl liefert, die dem Paar zugeordnet ist.
- Und einer, der zu einer natürlichen Zahl dasjenige Zahlenpaar liefert, das ihr zugeordnet ist.

Zahlenpaar \rightsquigarrow Zahl

input $k, m \in \mathbb{N}$;

output $n \in \mathbb{N}$;

$$n := k \cdot (k+3)/2 + k \cdot m + m \cdot (m+1)/2.$$

$\mathbb{N} \times \mathbb{N}$	0	1	2	...
0	(0, 0)	(0, 1)	(0, 2)	...
1	(1, 0)	(1, 1)	(1, 2)	...
2	(2, 0)	(2, 1)	(2, 2)	...
\vdots	\vdots	\vdots	\vdots	

Beispielsweise: $(k, m) = (1, 2) \rightsquigarrow n = 7$.

Zahl \rightsquigarrow Zahlenpaar

```
input  $n \in \mathbb{N}$ ;  
output  $(k, m) \in (\mathbb{N} \times \mathbb{N})$  (init  $(0, 0)$ );  
var  $i, s : \mathbb{N}$  (init 0);  
do  $i \neq n \rightarrow$  if  $(k, m) = (s, 0) \rightarrow s := s+1; (k, m) := (0, s)$ ;  
     $\square$   $(k, m) \neq (s, 0) \rightarrow (k, m) := (k+1, m-1)$   
    fi;  
     $i := i+1$   
od
```

\rightarrow lies: „dann“, \square lies: „oder falls“.

Beispielsweise: $n = 7 \rightsquigarrow (k, m) = (1, 2)$.

Elemente einer Programmiersprache

- Input/Output-Teil, z.B. **input** *Variable*.
- Deklaration **var** x : *Wertebereich*
- Leerkommando **skip**.
- Zuweisung $x := E$.
- Hintereinanderausführung zweier Kommandos $K_1; K_2$.
- Alternativkommando

if $B_1 \rightarrow K_1 \square B_2 \rightarrow K_2 \square \dots \square B_m \rightarrow K_m$ **fi**.

- Schleifenkommando

do $B_1 \rightarrow K_1 \square B_2 \rightarrow K_2 \square \dots \square B_m \rightarrow K_m$ **od**.

Beweis der Überabzählbarkeit von $\mathbb{N} \rightarrow \mathbb{N}$

Falls $\mathbb{N} \rightarrow \mathbb{N}$ abzählbar ist, gibt es eine surjektive Funktion $f: \mathbb{N} \rightarrow (\mathbb{N} \rightarrow \mathbb{N})$. Wir bilden eine Funktion $g: \mathbb{N} \rightarrow \mathbb{N}$ durch

$$g(x) = (f(x))(x) + 1 \text{ für alle } x \in \mathbb{N}.$$

Aus der Surjektivität von f folgt, dass es eine Zahl n mit $g = f(n)$ gibt. Jetzt gilt aber:

$$\begin{aligned} g(n) &= (\text{Definition von } g) \\ &\quad (f(n))(n) + 1 \\ &= (g = f(n)) \\ &\quad g(n) + 1. \end{aligned}$$

Dies ist ein Widerspruch! Unsere Annahme muss also falsch gewesen sein, und es gilt stattdessen: $\mathbb{N} \rightarrow \mathbb{N}$ ist überabzählbar.

Selbstanwendung / Diagonalisierung

Der Überabzählbarkeitsbeweis benutzt ein **Selbstanwendungs-** oder **Diagonalisierungsargument**, das im Kern auf G. Cantor zurückgeht. Er hat ein ähnliches Argument für den Nachweis der Überabzählbarkeit der Menge der reellen Zahlen zuerst verwendet.

	0	1	2	3	...
$f(0)$	$\neq g(0)$...
$f(1)$		$\neq g(1)$...
$f(2)$			$\neq g(2)$...
$g = f(3)$				$\neq g(3)$...
\vdots	\vdots	\vdots	\vdots		

Surjektivität: g ist z.B. $f(3)$. **Aber:** $g(3) = (f(3))(3) \neq g(3)$?!

Dieses Argument ist vom Diagonalverfahren des Abzählbarkeitsbeweises zu unterscheiden.

Alphabete

- Ein **Alphabet** ist eine **endliche, nicht leere** Menge von Grundsymbolen.
- Typische Bezeichnungen für Alphabete: Σ („**Symbole**“), Γ .
- Beispiele:

$$\Sigma_0 = \{0, 1\}$$

$$\Sigma_1 = \{a, b, c, d\}$$

$$\Sigma_2 = \{\sqcup, a, \dots, z, A, \dots, Z\} \quad (\sqcup = \text{Blankzeichen}).$$

Buchstaben

- Die Elemente eines Alphabets heißen **Buchstaben**.
- Typische Bezeichnungen für Buchstaben: a, b, \dots, S, X, \dots
- **Beispiele:**

Alphabet	Buchstaben
$\{0, 1\}$	0 und 1
$\{a, b, c, d\}$	a, b, c und d
$\{\sqcup, a, \dots, z, A, \dots, Z\}$	a, \dots, z, A, \dots, Z und \sqcup .

Wörter

- Ein **Wort über Σ** ist eine Aneinanderreihung von Buchstaben aus Σ . **Beispiele:** 0111 (Wort über Σ_0), *abc* (Wort über Σ_1), *Liebe* (Wort über Σ_2).
- Typische Bezeichnungen für Wörter: w, v, u, \dots
- Die **Länge** $|w|$ eines Wortes w ist die Anzahl der Buchstaben in ihm. **Beispiele:** $|0111| = 4$, $|abc| = 3$, $|Liebe| = 5$.
- Das **leere Wort** ε hat 0 Buchstaben: $|\varepsilon| = 0$.
- Jeder Buchstabe $a \in \Sigma$ ist auch ein Wort der Länge 1: $|a| = 1$.
- Die Menge aller Wörter über Σ wird mit Σ^* bezeichnet.
- Die Menge aller **nicht leeren** Wörter über Σ wird mit Σ^+ bezeichnet: $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$.

Abzählbarkeit von Σ^*

Satz: Σ^* ist abzählbar. Beweis?

Für $\Sigma = \{a, b, c\}$:

Σ^0	Wörter der Länge 0	$\{\varepsilon\}$
Σ^1	Wörter der Länge 1	$\{a, b, c\} (= \Sigma)$
Σ^2	Wörter der Länge 2	$\{aa, ab, ac, ba, bb, bc, ca, cb, cc\}$
Σ^3	Wörter der Länge 3	$\{aaa, aab, \dots, ccc\}$
\vdots		
Σ^n	Wörter der Länge n	$\{a^n, a^{n-1}b, \dots, c^n\}$

Allgemein gilt $|\Sigma^n| = |\Sigma|^n$.

Abzählung erst nach Länge, bei gleicher Länge (z.B.) **lexikalisch**
(wie oben oder wie im Wörterbuch).

Konkatenation

- Binäre Operation auf Wörtern.
- Seien $v = a_1 \dots a_n$ und $w = b_1 \dots b_m$:
$$v \cdot w = vw = a_1 \dots a_n b_1 \dots b_m.$$
- Sonderfälle $n = 0$: $v = \varepsilon$ und $m = 0$: $w = \varepsilon$.
- Es gilt $\varepsilon u = u = u\varepsilon$ für alle Wörter u .
- **Assoziativgesetz**: $u(vw) = (uv)w$ für alle Wörter u, v, w .
- n -fache Hintereinanderschreibung v^n eines Wortes v :
 $v^0 = \varepsilon$, $v^{n+1} = vv^n$ (induktive Definition).

Präfix, Suffix, Spiegelwort

- v ist **Teilwort** von w falls $\exists u, u' : uvu' = w$.
- v ist **Suffix** von w falls $\exists u : uv = w$.
- v ist **Präfix** von w falls $\exists u' : vu' = w$.
- **Beispiel**: 11 kommt in 1101110110 viermal als Teilwort, einmal als Präfix und keinmal als Suffix vor.
- Das zu v **reverse Wort** oder **Spiegelwort** v^R :
 $\varepsilon^R = \varepsilon$ und $(av)^R = v^R a$ (induktive Definition).
- **Beispiel**: $(Liebe)^R = ebeiL$

Sprachen

- Eine **Sprache** über einem Alphabet Σ ist eine Teilmenge $L \subseteq \Sigma^*$. Typische Bezeichnungen: L („**language**“), K , ...
- **Beispiele** für Sprachen über dem Alphabet $\Sigma_0 = \{0, 1\}$:

$$\begin{aligned} L_1 &= \emptyset && \text{die } \textit{leere Sprache} \\ L_2 &= \{\varepsilon\} && \text{die } \textit{Einheitssprache} \\ L_3 &= \{0, 1\}^* && = \{\varepsilon, 0, 1, 00, 01, \dots\}, \\ &&& \text{die } \textit{volle Sprache} \text{ über } \{0, 1\} \\ L_4 &= \{w0w \mid w \in \{1\}^*\} && = \{0, 101, 11011, \dots\} \\ L_5 &= \{ww^R \mid w \in \{0, 1\}^*\} && = \{\varepsilon, 00, 11, 0000, 0110, 1001, \dots\} \\ L_6 &= \{0^n 1^n \mid n \in \mathbb{N}\} && = \{\varepsilon, 01, 0011, 000111, \dots\} \\ L_7 &= \{(01)^n \mid n \in \mathbb{N}\} && = \{\varepsilon, 01, 0101, 010101, \dots\}. \end{aligned}$$

Charakteristische Funktion einer Sprache

- Die **charakteristische Funktion** χ_L einer Sprache $L \subseteq \Sigma^*$:

$$\chi_L: \begin{cases} \Sigma^* \rightarrow \{0, 1\} \\ w \mapsto \begin{cases} 1 & \text{falls } w \in L \\ 0 & \text{falls } w \notin L. \end{cases} \end{cases}$$

- χ_L ist **linkstotal** und **rechtseindeutig**,
also in der Tat eine **Funktion**.
- **Beispiel**: $L = \{0v \mid v \in \{0, 1\}^*\}$:

$$\chi_L(w) = \mathbf{if } w \text{ fängt mit } 0 \text{ an} \rightarrow 1 \quad \mathbf{else} \rightarrow 0 \mathbf{fi.}$$

- Weiteres **Beispiel**: es gilt $\chi_{L_1 \cup L_2}(w) = \max(\chi_{L_1}(w), \chi_{L_2}(w))$.

Partielle (halbe) char. Funktion einer Sprache

- Die **partielle** oder **halbe charakteristische Funktion** χ_L^+ einer Sprache $L \subseteq \Sigma^*$:

$$\chi_L^+ : \begin{cases} \Sigma^* \xrightarrow{p} \{1\} \\ w \mapsto \begin{cases} 1 & \text{falls } w \in L \\ \text{undef} & \text{falls } w \notin L. \end{cases} \end{cases}$$

- χ_L^+ ist **rechtseindeutig**, aber nicht notwendigerweise linkstotal, also in der Regel nur eine **partielle Funktion**.
- $f: X \xrightarrow{p} Y$ bedeutet: f bildet eine Teilmenge von X auf Y ab.
- **Definitionsbereich** $\text{dom}(f)$ von f , **Wertebereich** $\text{cod}(f)$ von f .

Operationen auf Sprachen

$L \cap K = \{w \mid w \in L \wedge w \in K\}$ (Durchschnitt von L und K)

$L \cup K = \{w \mid w \in L \vee w \in K\}$ (Vereinigung von L und K)

$L \setminus K = \{w \mid w \in L \wedge w \notin K\}$ (Differenz von L und K)

$\bar{L} = \Sigma^* \setminus L$ (Komplement von L)

$L \cdot K = \{uv \mid u \in L \wedge v \in K\}$ (Konkatenation von L und K)

$L^0 = \{\varepsilon\}, L^{n+1} = L \cdot L^n$ (n te Iterierte L^n von L)

$L^* = \bigcup_{n \in \mathbb{N}} L^n$ („Sternabschluss“ von L)

$L^+ = \bigcup_{n \in (\mathbb{N} \setminus \{0\})} L^n$ („Plusabschluss“ von L)

$L^R = \{w^R \mid w \in L\}$ (Spiegelsprache von L).

Mengentheoretische Sprachoperationen

- **Beispiel:** Seien $L_1 = \{1, 00\}$ und $L_2 = \{00, 01, 11\}$. Dann:

$$L_1 \cap L_2 = \{00\}$$

$$L_1 \cup L_2 = \{1, 00, 01, 11\}$$

$$L_1 \setminus L_2 = \{1\}$$

$$\overline{L_1} = \{0, 1\}^* \setminus \{1, 00\}.$$

- **de Morgan:** $\overline{L \cup K} = \overline{L} \cap \overline{K}$ und $\overline{L \cap K} = \overline{L} \cup \overline{K}$.

Sprachkonkatenationsoperationen

- **Beispiel:** Seien $L_1 = \{1, 00\}$ und $L_2 = \{00, 01, 11\}$. Dann:

$$L_1 L_2 = \{100, 101, 111, 0000, 0001, 0011\}$$

$$L_2 L_1 = \{001, 0000, 011, 0100, 111, 1100\}$$

$$L_1^3 = \{111, 1100, 1001, 10000, 0011, 00100, 00001, 000000\}$$

$$L_1^* = \{ \underbrace{\varepsilon}_{L_1^0}, \underbrace{1, 00}_{L_1^1}, \underbrace{11, 100, 001, 0000}_{L_1^2}, \underbrace{111, \dots}_{L_1^3}, \dots \}$$

$$L_1^+ = L_1^* \setminus \{\varepsilon\}.$$

- **Assoziativgesetz:** $(H \cdot K) \cdot L = H \cdot (K \cdot L).$

Spiegeln einer Sprache

- **Beispiel:** Seien $L_1 = \{1, 00\}$ und $L_2 = \{00, 01, 11\}$. Dann:

$$L_1^R = \{1, 00\} (= L_1)$$

$$L_2^R = \{00, 10, 11\} (\neq L_2)$$

$$L_1 L_2^R = \{100, 110, 111, 0000, 0010, 0011\}.$$

- **Spiegeln ist selbstinvers:** $(L^R)^R = L$.

Abgeschlossenheit gegenüber Operationen

- Sei \mathcal{L} eine Klasse (d.h.: Menge) von Sprachen.
- \mathcal{L} heißt **abgeschlossen** gegenüber einer Operation auf Sprachen, wenn mit den Argumenten der Operation auch immer das Ergebnis der Operation in \mathcal{L} liegt.
- **Beispiel:** \mathcal{L} heißt **abgeschlossen gegenüber der Konkatenation**, wenn aus $L \in \mathcal{L} \wedge K \in \mathcal{L}$ folgt: $L \cdot K \in \mathcal{L}$.
- **Konkretes Beispiel:** $\mathcal{E}\mathcal{N}\mathcal{D}\mathcal{L}$ sei die Klasse der endlichen Sprachen über Σ . Dann ist $\mathcal{E}\mathcal{N}\mathcal{D}\mathcal{L}$ abgeschlossen gegenüber der Konkatenation \cdot , aber nicht gegenüber dem Sternabschluss $*$.
- **NB:** Man spricht von einer **Klasse**, wenn es sich um eine **große** Menge handelt.