Martin-Luther-Universität Halle-Wittenberg
Institut für Informatik
Prof. Dr. L. Staiger, Dipl.- Math. Ralf Stiebe

Skript zur Vorlesung

Ausgewählte Kapitel der Theoretischen Informatik

Wintersemester 1996, Version vom 16.4.97

Inhaltsverzeichnis

1 Halbgruppen und Monoide **2**
1.1 Halbgruppen und Monoide . . . 2
1.2 Eigenschaften von Wörtern . . . 5
1.3 Automaten und Halbgruppen . . . 8

2 Codes **9**
2.1 Präfix-Codes und Bäume . . . 12

3 Sprachentheoretische Eigenschaften von Codes **17**
3.1 Der Satz von Sardinas und Petterson . . . 17
3.2 Maximalität und Vollständigkeit . . . 20
3.3 Decodierverzögerung . . . 24

4 Kryptologie **26**
4.1 Klassische Verfahren / Symmetrische Verfahren . . . 26
4.1.1 Monoalphabetische Verfahren . . . 26
4.1.2 Polyalphabetische Verfahren . . . 28
4.2 Asymmetrische Verfahren / Public Key Kryptographie . . . 30
4.2.1 Darstellung des Grundgedanken asymmetrischer Verfahren . . . 30
4.2.2 Zahlentheoretische Grundlagen . . . 30
4.3 Primzahltests . . . 33
4.4 RSA - Verfahren . . . 34
4.5 Elektronische Unterschriften . . . 35

5 Kolmogorov Komplexität **35**
5.1 Anwendungen der Kolmogorov Komplexität . . . 39

6 Kolmogorov Komplexität unendlicher Folgen **40**

7 EA auf unendlichen Wörtern **43**
7.1 Topologie in X^ω . . . 48

A Übungsaufgaben **51**

B Lösungen ausgewählter Aufgaben **60**

Literatur **61**

Vorwort

Einleitung

1 Halbgruppen und Monoide

1.1 Halbgruppen und Monoide

Definition 1.1 (Halbgruppe) $(H, \cdot)$ heißt Halbgruppe $:\Longleftrightarrow$

1. H ist nichtleere Menge (Trägermenge)
2. $\cdot : H \times H \to H$ ist eine zweistellige Operation
3. $\cdot$ ist assoziativ, d.h. $\forall x \forall y \forall z\,(x, y, z \in H \to (x \cdot y) \cdot z = x \cdot (y \cdot z))$

Definition 1.2 (Monoid) $(M, \cdot, \mathbb{1})$ heißt Monoid $:\Longleftrightarrow$

1. $(M, \cdot)$ ist Halbgruppe
2. $\mathbb{1}$ ist Einselement in $(M, \cdot)$, d.h. $\forall x(x \in M \to x \cdot \mathbb{1} = \mathbb{1} \cdot x = x)$

Definition 1.3 (a) U heißt Unterhalbgruppe von $(H, \cdot) :\Longleftrightarrow$

1. $U \subseteq H$
2. U ist abgeschlossen bzgl. $\cdot$, d.h. $\forall x \forall y(x, y \in U \to x \cdot y \in U)$

(b) U heißt Untermonoid von $(M, \cdot, \mathbb{1}) :\Longleftrightarrow$

1. U ist Unterhalbgruppe von $(M, \cdot)$
2. $\mathbb{1} \in U$

Bemerkung 1.1 Sei $(H, \cdot)$ eine Halbgruppe ohne Einselement. Setze $H^1 := H \cup \{\mathbb{1}\}$. Definiere für alle $x \in H : \mathbb{1} \cdot x = x \cdot \mathbb{1} = x$, $\mathbb{1} \cdot \mathbb{1} = \mathbb{1}$. Dadurch wurde $\cdot$ auf H^1 assoziativ fortgesetzt, d.h. $(H^1, \cdot, \mathbb{1})$ ist ein Monoid.

Definition 1.4 (a) Es sei $(H, \cdot)$ Halbgruppe und $B \subseteq H$. Dann heißt $B^+ := \{h_1 \cdots h_n, n \in \mathbb{N} \setminus \{0\}, h_i \in B\}$ die von B erzeugte Unterhalbgruppe von H.

(b) Sei $(M, \cdot, \mathbb{1})$ Monoid und $B \subseteq M$. Dann heißt $B^{*1} := \{h_1 \cdots h_n, n \in \mathbb{N}, h_i \in B\}$ das von B erzeugte Untermonoid von M.

Definition 1.5 (Erzeugendensystem) Sei H Halbgruppe. B heißt Erzeugendensystem von $H :\Longleftrightarrow$

$$\forall h(h \in H \to \exists m \exists h_1 \ldots \exists h_m\,(m \in \mathbb{N} \wedge h_i \in B \wedge h = h_1 \cdots h_m))$$

Definition 1.6 (Homomorphismus) (a) Seien $(H, \cdot)$, $(H', \cdot)$ Halbgruppen. Eine Abbildung $\varphi : H \to H'$ heißt Homomorphismus $:\Longleftrightarrow$ $\forall x, y \in H : \varphi(x \cdot y) = \varphi(x) \cdot \varphi(y)$

(b) Seien $(M, \cdot, \mathbb{1})$, $(M', \cdot, \mathbb{1}')$ Monoide. Eine Abbildung $\varphi : M \to M'$ heißt Homomorphismus : $\Longleftrightarrow$

- φ ist Halbgruppenhomomorphismus von $(M, \cdot)$ nach $(M', \cdot)$
- $\varphi(\mathbb{1}) = \mathbb{1}'$

Beispiel 1.1 Sei $M = D(\mathbb{N}^{2\times 2})$ Menge der 2×2-Diagonalmatrizen über $\mathbb{N}$. M ist mit Matrizenmultiplikation und $\mathbb{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ ein Monoid. Definiere Abbildung φ durch

$$\varphi\left(\begin{pmatrix} x & 0 \\ 0 & y \end{pmatrix}\right) = \begin{pmatrix} x & 0 \\ 0 & 0 \end{pmatrix}$$

Dann ist φ Halbgruppen- aber nicht Monoidhomomorphismus.

Satz 1.1 *Es sei $\varphi : H_1 \to H_2$ ein Halbgruppenhomomorphismus. Dann ist die Relation $R = \{(h, h') : h, h' \in H_1 \wedge \varphi(h) = \varphi(h')\}$ eine Kongruenzrelation auf H_1.*

Bemerkung 1.2 R ist Kongruenzrelation auf H_1, falls aus $(h, h'), (g, g') \in R$ stets $(hg, h'g') \in R$ folgt und R eine Äquivalenzrelation ist.

Beweis.(Satz 1.1)

(R) Da $\varphi(x) = \varphi(x)$ folgt $(x, x) \in R$, d.h. die Relation ist reflexiv.

(S) Sei $(x, y) \in R$, d.h. $\varphi(x) = \varphi(y)$, somit ist $\varphi(y) = \varphi(x)$ und folglich $(y, x) \in R$, d.h. die Relation ist symmetrisch.

(T) Sei $(x, y), (y, z) \in R$, d.h. $\varphi(x) = \varphi(y)$ bzw. $\varphi(y) = \varphi(z)$, und folglich $\varphi(x) = \varphi(z)$ bzw. $(x, z) \in R$, d.h. die Relation ist transitiv.

(K) Sei $(x, x'), (y, y') \in R \to \varphi(x) = \varphi(x'), \varphi(y) = \varphi(y')$. Da φ Halbgruppenhomomorphismus folgt $\varphi(xy) = \varphi(x)\varphi(y) = \varphi(x')\varphi(y') = \varphi(x'y') \to \varphi(xy) = \varphi(x'y') \to (xy, x'y') \in R$, d.h. die Relation respektiert die Operation.

□

Folgerung 1.2 *Es sei $\varphi : H_1 \to H_2$ ein Halbgruppenhomomorphismus und $B \subseteq H_1$ ein Erzeugendensystem von H_1. Dann ist φ schon durch die Werte über B eindeutig bestimmt.*

Beweis. Da B Erzeugendensystem von H_1 gilt : $\forall h \in H_1 \exists m \in \mathbb{N}$ und $h_1 \dots h_m \in H_1$ mit $h = h_1 \cdots h_m$. Da φ Homomorphismus ist, gilt $\varphi(h) = \varphi(h_1) \cdots \varphi(h_m)$, da die Werte $\varphi(h_1) \dots \varphi(h_m)$ bekannt sind, folgt $\varphi(h)$ ist bekannt. □

Bemerkung 1.3 Die Werte von φ auf B sind nicht frei, d.h. unabhängig voneinander wählbar.

Definition 1.7 (freies Erzeugendensystem) Eine Teilmenge $B \subseteq H$ einer Halbgruppe H heißt freies Erzeugendensystem von H : $\Longleftrightarrow$

1. B ist Erzeugendensystem

2. Und gilt $\forall_{\nu=1}^{m} a_\nu$ und $\forall_{\nu=1}^{k} b_\nu, a_\nu, b_\nu \in B$ mit $\prod_{\nu=1}^{m} a_\nu = \prod_{\nu=1}^{k} b_\nu$, dann gilt $m = k$ und $a_\nu = b_\nu \quad \forall \nu$.

H heißt freie Halbgruppe $\iff$ H besitzt ein freies Erzeugendensystem.

Beispiel 1.2 (a) $(\mathbb{N}, +)$ ist frei mit $B = \{1\}$

(b) $(\mathbb{N} \setminus \{0\}, \cdot)$ ist nicht frei da $2 \cdot 3 = 3 \cdot 2$

(c) $(X, \cdot)$ Worthalbgruppe über dem Alphabet X ist frei

Bemerkung 1.4 - Die Eindeutigkeit der Darstellung wird nur für endliche Wörter gefordert, da nur Sprachen mit endlichen Wörtern betrachtet werden. Entscheidende Bedeutung gewinnt dies z.B. im Beweis des Satzes über die Existenz maximaler Codes.

- Eine freie nichttriviale Halbgruppe ist somit immer nichtkommutativ.

- Ein freies Erzeugendensystem ist minimal und das eventuell vorhandene Element $\mathbb{1}$ gehört nicht zu diesem System, da $\mathbb{1} \cdot \mathbb{1} = \mathbb{1}$.

- Eine endliche nichttriviale Halbgruppe ist niemals frei.

Definition 1.8 Eine Abbildung $\lambda : H \to \mathbb{N}$ heißt Länge : $\iff$

$$\lambda(h \cdot h') = \lambda(h) + \lambda(h')$$

(d.h. λ ist Homomorphismus von $(H, \cdot)$ nach $(\mathbb{N}, +)$). λ heißt strikte Länge : $\iff$ $\lambda^{-1}(0) \subseteq \{\mathbb{1}\}$. Schreibweise: $|h| := \lambda(h)$.

Bemerkung 1.5 Sei M Monoid mit strikter Länge. Dann ist $M \setminus \{\mathbb{1}\}$ eine Halbgruppe.

Beweis. Sei $h, h' \in M \setminus \{\mathbb{1}\}$. Dann ist $h \cdot h' \neq \mathbb{1}$, da $|h \cdot h'| = |h| + |h'| > 0$. □

Definition 1.9 (a) Ein Element h einer Halbgruppe H heißt unzerlegbar, falls es kein $h', h'' \in H$ mit $h = h'h''$ gibt.

(b) Ein Element h eines Monoids M heißt unzerlegbar, falls $h = h'h''$ stets $h' = \mathbb{1}$ oder $h'' = \mathbb{1}$ impliziert.

Folgerung 1.3 *Es sei H eine Halbgruppe ohne Einselement. Dann ist $H \setminus H^2$ die Menge der unzerlegbaren Elemente von H, mit $H^2 = \{h'h'' : h', h'' \in H\}$.*

Beweis. folgt aus Definition 1.9. □

Satz 1.4 *Jede Halbgruppe H ohne Einselement hat ein eindeutig bestimmtes minimales Erzeugendensystem B, falls H eine Halbgruppe mit strikter Länge ist, gilt $B = H \setminus H^2$.*

Beweis.

- $B^+ \subseteq H$, bei Definition von B^+
- $B^+ \supseteq H$ Induktion über die Länge

IA: $h \in H$ mit minimaler Länge $\rightarrow h$ unzerlegbar $\rightarrow h \in B \subseteq B^+$

IV: Für alle h mit $|h| \leq n$ gelte $h \in B^+$.

IB: $|h| = n + 1 \rightarrow h \in B^+$

IS:
- Sei $|h| = n + 1$, zeige $h \in B^+$
- h unzerlegbar $\rightarrow h \in B \subseteq B^+$, d.h Satz ist wahr
- $h = h_1 h_2$ mit $h_1, h_2 \in H \rightarrow |h_1|, |h_2| \leq n \rightarrow h_1, h_2 \in B^+$ (Induktionsvoraussetzung) $\rightarrow h_1 h_2 \in B^+$

B ist minimal, weil alle $h \in B$ unzerlegbar.

□

Definition 1.10 Setze

$$\sqrt{M} := (M \setminus \{\mathbb{1}\}) \setminus (M \setminus \{\mathbb{1}\})^2$$

und

$$\sqrt{H^1} := \sqrt{H \cup \{\mathbb{1}\}} = H \setminus H^2,$$

falls H Halbgruppe ohne Einselement ist.

Satz 1.5 *Jedes Monoid M mit strikter Länge hat $\sqrt{M}$ als minimales Erzeugendensystem.*

Beweis. folgt aus Satz 1.4. □

Satz 1.6 (Lévi, 1944) *Es sei M ein Monoid. M ist genau dann frei, wenn die folgenden Bedingungen gelten*

1. *M hat strikte Länge*
2. *Falls $w_1 w_2 = w_3 w_4$, wobei $w_i \in M$, so gilt*

$$\exists w_5 \in M : w_1 = w_3 w_5, \quad w_4 = w_5 w_2$$

oder

$$\exists w_6 \in M : w_3 = w_1 w_6, \quad w_2 = w_6 w_4$$

Beweis. Zu kompliziert zum TEXen, jedoch eigentlich trivial. □

1.2 Eigenschaften von Wörtern

Im folgenden betrachten wir die freie Worthalbgruppe X^* über dem gegebenen Alphabet X, welches mit dem leeren Wort e zu einem Monoid erweitert werden kann. Die Operation ist das Aneinanderhängen.

Folgerung 1.7 *Es seien $u, v, u', v' \in X^*, |u| \leq |u'|$ und $uv = u'v' \rightarrow \exists w \in X^*$ mit $uw = u'$ und $wv' = v$ (Satz von Lévi).*

Lemma 1.8 Es seien $u, v, w \in X^*, uv = vw$und $u, v \neq e$. Dann gibt es $s, t \in X^*$ und ein $k \in \mathbb{N}$ mit $u = st, \quad v = (st)^k, \quad w = ts$.

Beweis. Der Beweis wird mittels Induktion über die Länge von v geführt.

(IA) $|v| = 1$, und damit $|v| \leq |u|$ in diesem Falle setze $s = v$ und bestimme t durch $u = vt$, d.h. $w = ts, k = 0$

(IV) Behauptung sei wahr für alle Paare $u, v \in X^*$ mit $|v| \leq n$

(IB) Behauptung ist wahr für alle Paare $u, v \in X^*$ mit $|v| = n + 1$

(IS) Sei $uv = vw \rightarrow \exists w' \in X^*$ mit $uw' = v, w'w = v$ (Satz von Lévi) $\rightarrow w'w = uw'$, da $|u| \geq 1$ folgt somit $|w'| \leq n$, d.h. für $u, w', w \in X^* \exists k \in \mathbb{N}$ und $\exists s', t' \in X^*$ mit $u = s't', w = t's', w' = (s't')^k s'$ (Induktionsvoraussetzung) $\rightarrow v = w'w = (s't')^{k+1}s'$

□

Lemma 1.9 Es seien $u, v \in X^*$ mit $uv = vu$. Dann existiert ein $s \in X^*$ mit $u = s^m, v = s^n$ für geeignet zu wählende $m, n \in \mathbb{N}$.

Beweis. Der Beweis erfolgt diesmal durch Induktion über die Länge des durch uv gebildeten Wortes.

(IA) $|uv| = 0 \rightarrow s = e, m = n = 1$

(IV) Behauptung gelte für alle Paare u, v mit $|uv| \leq n$

(IB) Behauptung gelte für alle Paare u, v mit $|uv| = n$

(IS) Sei $uv = vu, |uv| = n + 1 \rightarrow \exists s, t \in X^*, k \in \mathbb{N}$ mit $u = st, v = (st)^k s$, wegen $uv = vu$ folgt $(st)^{k+1}s = (st)^k sst \rightarrow ts = st$, da $u = st$ folgt $|ts| < |uv|$ für $v \neq e$, Behauptung folgt nun aus IV, da $t = a^k, s = a^l$ für ein gewisses $a \in X^*$ und für $k, l \in \mathbb{N}$. Und da für $v = e, s = u, m = n = 1$ gesetzt werden kann, ist das Lemma bewiesen.

□

Lemma 1.10 Es seien $u, v \in X^*$, so gilt $uv = vu \iff$ es gibt ein $w \in X^*$ mit

1. $|w| = |u| + |v|$
2. $w \sqsubseteq u^m$
3. $w \sqsubseteq v^n$

für geeignetes $m, n \in \mathbb{N}$.

Bemerkung 1.6 Schreibe $w \sqsubseteq w' \iff \exists t \in X^*$ mit $wt = w'$, d.h. w ist Präfix von w'

Beweis.

- Es sei $uv = vu$, dann $u = s^m, v = s^n$ für geeignetes $m, n \in \mathbb{N}, s \in X^* \rightarrow u^n = v^m = s^{mn} \rightarrow$ Behauptung $1, 2, 3$

- Betrachte vu^m, v^{n+1}, wir wissen $w \sqsubseteq v^n, v^{n+1}, u^m \rightarrow w \sqsubseteq vu^m \rightarrow w = vu$ da $|w| = |vu|$, analog zeige $w = uv \rightarrow uv = vu$

□

Folgerung 1.11 *Ist $uv = vu$, so gibt es $m, n \in \mathbb{N}$ mit $u^m = v^n$. Man findet im obigen Beweis des letzten Lemmas den Beweis zu dieser Behauptung.*

Definition 1.11 $w \in X^*$ heißt primitives Wort $\iff$ aus $w = u^n$ stets $u = w$ und $n = 1$ folgt.

Bemerkung 1.7 Es sollte insbesondere $w \neq e$ beachtet werden.

Beispiel 1.3

- ab mit $a, b \in X$ und $a \neq b$
- abc, abb, aab mit $a, b, c \in X$ und a, b, c paarweise verschieden
- Worte mit Primzahllänge und $w \neq a^{|w|}, \quad a \in X$

Satz 1.12 *Ist $u \neq e$, so gibt es ein eindeutig bestimmtes primitives Wort w mit $u = w^n$, mit $n = \frac{|u|}{|w|}$.*

Beweis. Angenommen wir hätten zwei Darstellungen $u = w^k = v^l$, mit w, v primitiv. Dann gilt mit dem letzten Lemma $wv = vw$, da wir $w = u^{2k} = v^{2l}$ setzen können. Also $\exists \in X^*$ mit $w = s^k, v = s^r \rightarrow s = w = v, k = r = 1$ (v, w primitive Wörter). □

Definition 1.12 Sei $w \in X^*$ und $w = p^n$ mit p primitiv, dann setze $\sqrt{w} = p$. Die Definition ist korrekt, da p eindeutig durch w bestimmt.

Bemerkung 1.8 (Anwendung von primitiven Wörtern) Sei ein Punkt in der Ebene gegeben mit n Nachbarpunkten ohne Kompaß, d.h. Nachbarschaft ist bzgl. Drehung nicht unterscheidbar. Die Punkte der Nachbarschaft seien eventuell unterschiedlich gefärbt, bei Verwendung von r Farben. Wieviele unterscheidbare Nachbarschaften gibt es ? Andere Formulierung : Die Frage ist äquivalent zu der Frage nach der Anzahl von zirkulären Wörtern mit Länge n über einem Alphabet der Größe r. Lösung : Sei ξ zirkuläres Wort der Länge n, schneide ξ an einer beliebigen Stelle s_0 auf. $\rightarrow$ 'normales' Wort w der Länge n. Es gibt $l = \frac{n}{|\sqrt{w}|}$ verschiedene Schnittstellen, so daß das entstehende Wort mit w übereinstimmt, ($w = u^l, u = \sqrt{w}$, mit u). Sei Q_n gegeben durch die Anzahl der primitiven Wörter der Länge n, so gilt

- $Q_p = r^p - 1$, p Primzahl, da jedes Wort mit Primzahllänge primitiv ist, abgesehen von $w = a^p, a \in X$.
- $r^n = \sum_{d|n} Q_d$, da jedes Wort der Länge n eindeutig durch seine primitive Wurzel bestimmt ist.

Sei Z_n gegeben durch die Anzahl der zirkulären Wörter der Länge n, so gilt $Z_n = \sum_{d|n} \frac{Q_d}{d}$, (Die Schnittstelle kann auch in einem primitiven Wort sein.) Die Berechnung der Z_n kann rekursiv erfolgen. Es ist möglich eine explizite Summendarstellung für die Q_n anzugeben mittels der Möbiusschen Inversionsformel (Zahlentheorie).

Satz 1.13 *Es sei $u \in X^*, u \neq a^{|u|}, a \in X$, dann ist wenigstens eines der beiden Wörter u, ua primitiv.*

Beweis. Angenommen der Satz ist falsch, d.h. es gibt ein $u \in X^*$ und ein $a \in X$ mit folgenden Eigenschaften: $u = v^k, ua = w^l, |v|, |w| \neq 0, k, l > 1$, mit u, v primitiv. Wir wissen bereits $u \sqsubseteq v^k \sqsubseteq w^l$, weiterhin gilt $|w| + |v| = \frac{|u|+1}{l} + \frac{|u|}{k} \leq |u| * (l^{-1} + k^{-1}) + l^{-1} < |u| + 1 \rightarrow |w| + |v| \leq |u|$, nach Anwendung eines der vorangegangenen Lemma können wir sagen : $\exists m, n \in \mathbb{N} \exists s \in X^*$ mit $w = s^m, v = s^n \rightarrow u = s^{nk}, ua = s^{lm} \rightarrow s = a$, d.h. $u = a^{|u|}$, dies steht jedoch im Widerspruch zur Voraussetzung. □

Bemerkung 1.9 Eine Folgerung des Satzes ist die Existenz eines primitiven Wortes zu vorgegebener Länge $n \in \mathbb{N}$. Es gilt jedoch sogar das Folgende : Die Anzahl der primitiven Wörter mit Länge n ist größer als die Hälfte der Anzahl aller Wörter mit Länge n. Beweis erfolgt mittels Induktion unter Verwendung der Formel für die Anzahl der primitiven Wörter.

1.3 Automaten und Halbgruppen

Im folgenden wird das durch $(A^A, \bullet, id_A)$ gegebene Monoid betrachtet, wobei

- $A^A := \{f | f : A \rightarrow A\}$, d.h. Menge aller Abbildungen von A auf sich.
- id_A die Identität auf A
- $f \bullet g(x) = g(f(x))$

Die Operation $\bullet$ ist assoziativ.

Satz 1.14 *Ist M ein freies Monoid, B freies Erzeugendensystem von M, M' Monoid, $\varphi : B \rightarrow M'$, dann läßt sich φ auf eindeutige Weise zu einem Monoidhomomorphismus von M in M' fortsetzen.*

Beweis. Sei $v \in M$, da B freies Erzeugendensystem folgt $v = \prod_{\nu=1}^{n} b_\nu$, setze $\varphi(v) = \prod_{\nu=1}^{n} \varphi(b_\nu)$. Da die Darstellung von v über B eindeutig ist, folgt Korrektheit der Definition von φ. Die Homomorphismuseigenschaft von φ ist ebenfalls mit obiger Definition gesichert. □

Bemerkung 1.10 M freies Monoid $\rightarrow B$ eindeutig bestimmt.

Beispiel 1.4 Die Voraussetzung eines freien Erzeugendensystem ist essentiell. Setze $M = M' = X^*, X = \{a, b\}, B = \{a, aa, b\}$. Die folgende Funktion φ gegeben über B durch $\varphi(a) = a, \varphi(aa) = b, \varphi(b) = a$ läßt sich offenbar nicht als Homomorphismus auf ganz M fortsetzen, da sonst $aa = \varphi(a)\varphi(b) = \varphi(aa) = b$ gelten müßte, aber $b \neq aa$.

Definition 1.13 (Transitionsmonoid) Sei $\mathbb{A}$ ein endlicher Automat gegeben durch $\mathbb{A} = (X, Z, z_0, f, F)$ mit $z_0 \in Z, F \subseteq Z, f : Z \times X \rightarrow Z$, wobei $|X|, |Z| < \infty$. Sei $x \in X$, sei f_x definiert durch $f_x : Z \rightarrow Z, f_x(z) = f(z, x)$. Die Abbildung $\varphi_A : X \rightarrow Z^Z$ mit $\varphi(x) = f_x$ läßt sich nach dem obigen Satz zu einem Monoidhomomorphismus zwischen $(X^*, \cdot, e)$ und $(Z^Z, \bullet, id_Z)$ fortsetzen. Bezeichne als Transitionsmonoid das durch $(\varphi_A(X^*), \bullet, id_Z)$ gegebenen Monoid.

Bemerkung 1.11 - vereinfachende Schreibweise : $f(z, x_1x_2 \dots x_n) = f_{x_1} \bullet f_{x_2} \bullet \dots \bullet f_{x_n}(z)$

- $\varphi_A(e) = f_e = id_Z$

Satz 1.15 (Kleene, Myhill, Nerode) *Es sei $L \subseteq X^*$. L ist reguläre Sprache $\iff$ es ein endliches Monoid $(M, \cdot, 1)$, $P \subseteq M$, und einen Homomorphismus $\varphi : X^* \to M$ mit der Eigenschaft $L = \varphi^{-1}(P)$ gibt.*

Definition 1.14 (Nerode Relation und syntaktische Relation) Die durch $w \sim_L v \iff (\forall s \in X^*$ gilt $ws \in L \iff vs \in L)$ definierte Relation heißt Nerode Relation. Die durch $w \simeq_L v \iff (\forall t, s \in X^*$ gilt $tws \in L \iff tvs \in L)$ definierte Relation heißt syntaktische Relation.

Bemerkung 1.12 In Theoretischer Informatik 1 wurde gezeigt :

- Nerode-Relation ist Äquivalenzrelation
- L reguläre Sprache $\iff$ Nerode-Relation $\sim_L$ von endlichen Index.

Der Beweis, daß $\simeq_L$ eine Äquivalenzrelation ist, ist analog zum Beweis für $\sim_L$.

Beweis. L reguläre Sprache $\to \exists\, DEA\, A = (X, Z, z_0, f, F)$ mit $L = \{w | f(z_0, w) \in F\} = \{w | f_w(z_0) \in F\}$ Betrachtung des Monoids Z^Z mit $Z^Z = (Z^Z, \bullet, id_Z)$ sowie der durch $P = \{f | f \in Z^Z$ und $f(z_0) \in F\}$ definierten Teilmenge von Z^Z. Als $\varphi : X^* \to Z^Z$ wird derjenige Homomorphismus verwendet, der in der Definition des zu A gehörenden Transitionsmonoid definiert bzw. angegeben wurde. Es reicht nun aus zu zeigen, daß $\sim_L$ endlichen Index hat. Man hat $w \simeq_L v \to w \sim_L v\, (t = e)$, d.h. $\simeq_L$ hat mindestens genauso viele Äquivalenzklassen wie $\sim_L$, m.a.W. Index $\simeq_L \geq$ Index $\sim_L$. Damit reicht es aus zu zeigen $\simeq_L$ hat endlichen Index. Wir zeigen aus $\varphi(w) = \varphi(v) \to w \simeq v$, d.h. $|\varphi(X^*)| \geq$ Index $\simeq_L$ und da $\varphi : X^* \to Z^Z, Z^Z$ endlicher Monoid würde die Behauptung folgen. $\varphi(w) = \varphi(v) \to \varphi(swt) = \varphi(s)\varphi(w)\varphi(t) = \varphi(s)\varphi(v)\varphi(t) = \varphi(svt)$ $swt \in L \iff \varphi(swt) \in P \iff \varphi(svt) \in P \iff svt \in L$ □

2 Codes

Die Theorie der Codes teilt sich auf in

- Kryptologie
- Datenübertragung über gestörte Kanäle (Informationstheorie)
- Quellenkodierung, Datenkompression (Informationstheorie)

In allen drei Bereichen trifft man auf das folgende Schema :
Sender → Codierer → Kanal → Decodierer → Empfänger.
Dabei kann der Codierer als eine Funktion $\varphi : Y \to X^*$ und der Decodierer als eine weitere Funktion $\psi : X^* \to Y^*$ aufgefaßt werden, wobei die Art des Kanals das Alphabet X festlegt.

Definition 2.1 (Code, eindeutig decodierbarer Code, und Präfix-Code)
Die durch $C = \varphi(Y) \subseteq X^*$ definierte Teilmenge von X^* heißt Code. Der Code C heißt eindeutig decodierbarer Code $\iff$ C ist ein freies erzeugenden System von C^*. Im weiteren Verlauf wird nur äußerst selten zwischen einem allgemeinen Code und einem eindeutig decodierbaren Code unterschieden, d.h. i.a. ist ein Code eindeutig decodierbar. Der Code C heißt Präfix-Code bzw. präfixfreier Code : $\iff$ Es existieren keine $w_1, w_2 \in C$ mit w_1 ist Präfix von w_2. Analoge Definition für Suffix Code.

Beispiel 2.1 (a) Zählcode (längenvariabler Code)

$L0 \to 1$

$LL0 \to 2$

$\dots$

$LLLLLLLLL0 \to 10$

(b) 1 aus 10 Code (fixierte Länge)

$000000000L \to 1$

$00000000L0 \to 2$

$\dots$

$L000000000 \to 10$

(c) 2 aus 10 Code (fixierte Länge), analog zu 1 aus 10 Code

(d) Morsecode (längenvariabler Code)

Weitere Beispiele, sowie die Definition von Decodierverzögerung N später, falls ich die Zeit dazu finde.

Folgerung 2.1 *Jeder Präfix-Code ist eindeutig decodierbar. Ist C ein eindeutig decodierbarer Code, so ist $(C^*, \cdot, e)$ ein freies Untermonoid von $(X^*, \cdot, e)$. Ist C eindeutig decodierbarer Code, so ist $e \notin C$, da C freies Erzeugendensystem von C^*.*

Folgerung 2.2 *Ist $(M, \cdot, e)$ ein freies Untermonoid von $(X^*, \cdot, e)$, so ist $\sqrt{M}$ ein eindeutig decodierbarer Code für geeignet zu wählendes Alphabet Y. Zum Beweis siehe Satz von Lévi.*

Satz 2.3 (Zur Existenz eines Präfix-Codes) *Es sei $(l_i)_{i\in I}$ eine Folge natürlicher Zahlen, wobei nicht notwendigerweise $l_i \neq l_j, i \neq j$ erfüllt sein muß, die der Bedingung $\sum_{i\in I} r^{-l_i} \leq 1$ mit $r \in \mathbb{N}, r > 1$ genügt. Dann gibt es für X mit $|X| = r$ einen Präfix-Code derart, daß falls $C = \{w_i\}_{i\in I}$ die Bedingung $|w_i| = l_i \forall\, i \in I$ erfüllt ist. Die in der Voraussetzung aufgetauchte Ungleichung, wird als Ungleichung von Kraft bezeichnet.*

Definition 2.2 (lexiko- und quasilexikographische Ordnungsrelation)

(a) Sei $X = \{1, \dots, r\}$ und $w_1 = a_1 \dots a_n, w_2 = b_1 \dots b_m \in X^*$, so sage $a \leq b \iff a = b$ oder es gibt $i \in \mathbb{N}$ mit $a_i < b_i$ und $a_j = b_j \quad \forall j < i$. Die so definierte Ordnung heißt lexikographische Ordnungsrelation.

(b) Sei $X = \{1, \ldots, r\}$ und $w_1 = a_1 \ldots a_n, w_2 = b_1 \ldots b_m \in X^*$, so sage $a \leq b \iff a = b$ oder $n < m$ oder falls $n = m$, es gibt $i \in \mathbb{N}$ mit $a_i < b_i$ und $a_j = b_j \quad \forall j < i$. Die so definierte Ordnungsrelation heißt quasilexikographische Ordnungsrelation.

Beispiel 2.2 $X = \{1, 2\} \rightarrow 111 < 12$ lexikographisch, aber $12 < 111$ quasilexikographisch.

Bemerkung 2.1 Zu beweisen wäre, daß beide Relationen Ordnungsrelationen sind, d.h. die Bedingungen (R), (AS), (T) erfüllen.

(R) $a \leq a$

(AS) $a \leq b, b \leq a \rightarrow a = b$

(T) $a \leq b, b \leq c \rightarrow a \leq c$

Dies zu tun ist jedoch nur Schreibarbeit. Eine weitere Bemerkung zur lexikographischen Ordnungsrelation : Jede Ordnungsrelation über einer abzählbaren Menge M kann ordnungsisomorph in die lexikographische Ordnungsrelation über $\mathbb{N}^*$ eingebettet werden, d.h. es gibt $\varphi : M^* \rightarrow \mathbb{N}^*$ mit $\varphi(x) \leq \varphi(y) \iff x \leq y$.

Beweis. Sei $I = \mathbb{N}$ oder $\exists n \in \mathbb{N}$ mit $I = \{1 \ldots n\}$ und sei o.B.d.A $l_1 \leq l_2 \leq \ldots$. Setze $w_1 = 1 \ldots 1$ mit $|w_1| = l_1$, $M_1 = X^* \backslash w_1 X^*$, $M_i = M_{i-1} \backslash w_i X^*$ mit w_i erstes Wort der Länge l_i in der lexikographischen Ordnung in M_{i-1}. Es gilt $w_i \not\sqsubseteq w_j$ und $w_j \not\sqsubseteq w_i$ da $|w_i| \leq |w_j|$ und $w_j \notin w_i X^*$ mit $(i \leq j)$. Zu zeigen bleibt $M_{i-1} \cap X^{l_i} \neq \emptyset$, wobei X^{l_i} Menge der Wörter der Länge l_i. Anzahl der Wörter mit Länge l_i in $S = \{w_1 \ldots w_{i-1}\} X^* = \cup_{j=1}^{i-1} w_j X^*$ ist gegeben durch $|w_1 X^* \cap X^{l_i}| = r^{l_i - l_1}, \ldots, |w_{i-1} X^* \cap X^{l_i}| = r^{l_i - l_{i-1}} \rightarrow$ die gesuchte Anzahl für die Menge S beträgt $\sum_{k=1}^{i-1} r^{l_i - l_k} = r^{l_i} \sum_{k=1}^{i-1} r^{-l_k} < r^{l_i}$ (Ungleichung von Kraft ist gültig per Voraussetzung). Da $M_{i-1} = X^* \setminus S$ und da $|X^{l_i}| = r^{l_i}$ folgt $M_{i-1} \cap X^{l_i} \neq \emptyset$, d.h. die Existenz des Wortes w_i, d.h. jeder Konstruktionsschritt ist möglich. □

Definition 2.3 (Bernoulli - Maße auf X^*) Sei $\mu : X^* \rightarrow [0, 1]$ und erfülle die folgenden Bedingungen

- $\mu(X) = \sum_{w \in X} \mu(w) = 1$
- $\mu(w_1 w_2) = \mu(w_1)\mu(w_2) \forall w_1, w_2 \in X^*$
- $\mu(e) = 1$

So nenne μ ein Bernoulli - Maß (auf X^*). Sei $L \subseteq X^*, \mu(L) = \sum_{w \in L} \mu(w)$

Satz 2.4 (Satz von Mac Millan) *Sei μ ein Bernoulli - Maß, so gilt für jeden eindeutig decodierbaren Code $C \subseteq X^* \mu(C) \leq 1$.*

Folgerung 2.5 *Setze $\mu(w) = r^{-|w|}, r = |X|$, so ist μ ein Bernoulli - Maß und mit dem Satz von Mac Millan gilt für jeden eindeutigen Code die Ungleichung von Kraft.*

Beweis. Es ist ausreichend die Behauptung für endliche Codes nachzuweisen, da $C = \cup_{i=1}^{\infty} C_i$ mit $C_i = \{w_n \in C | n \leq i\}$, d.h. falls gültig für endliche Codes, d.h. $\mu(C_i) \leq 1 \rightarrow \mu(C) \leq 1$. (Beweis in Grundvorlesung Analysis) Sei C endlicher Code, setze $m = \max\{w_i | w_i \in C\} \rightarrow C \subseteq X^m$ Da C eindeutiger Code, d.h. ein freies Erzeugendensystem von C^* folgt $\mu(C^i) = \mu(C)^i \forall i \in \mathbb{N}$, da $S = \mu(\{w_1 \dots w_i | w_k \in C\}) = \sum_{w_1,\dots,w_i \in C} \mu(w_1) \dots \mu(w_i) = (\sum_{w \in C} \mu(w))^i = (\mu(C))^i$ dies folgt da jedes $w \in C^i$ eine eindeutige Darstellung bzw. Zerlegung über C besitzt per Definition und nach Anwendung von des Distributivgesetzes. Angenommen $\mu(C) = 1 + \epsilon, \epsilon > 0 \rightarrow \mu(C^i) = (1+\epsilon)^i$, andererseits $\mu(C^i) \leq \mu(X^{m*i}) = m * i + 1 \rightarrow (1+\epsilon)^i \leq m * i + 1 \forall i \in \mathbb{N} \rightarrow 1 + \epsilon \leq 1 \rightarrow \epsilon \leq 0$, steht im Widerspruch zur Definition von ϵ. □

2.1 Präfix-Codes und Bäume

Die nun folgende Theorie ist ein kleiner Einblick in das Gebiet der Informationstheorie, begründet von dem Nichtmathematiker Shannon, aber nichtsdestotrotz ist die Informationstheorie eines der schwersten Teilgebiete in der Mathematik. Sei B mathematischer Baum mit Wurzel w und r sei der maximaler Verzweigungsgrad von B, dann kann man B in X^* mit $|X| = r$ einbetten, wobei die Wurzel des Baumes dem Wort e, bestimmte $w \in X^*$ den Knoten und (w, wa) den Kanten des Baumes zugeordnet sind. Sei B ein vollständiger Baum der Höhe h, so tritt jedes Wort $w \in \cup_{i=0}^{h} X^i$ als Knoten des Baumes auf. $M \subseteq X^*$ ist Bild einer Einbettung eines Baumes $B \iff M$ ist präfix abgeschlossene Teilmenge von X^*,d.h. $v \in M, w \sqsubseteq v \rightarrow w \in M$

Folgerung 2.6 *Entspricht $B \subseteq X^*, B \neq \{e\}$ einem Baum, so bildet die Menge aller Blätter von B einen Präfix-Code. Ist umgekehrt $C \subseteq X^*$ ein Präfix-Code, so entspricht $B = \{w | w \in X^*, \exists v \in C, w \sqsubseteq v\}$, einem Baum mit den entsprechenden Knoten und Kanten, und die Menge der Blätter von B ist C.*

Da man Bäume als Speichermedien verwenden kann, die Zugriffshäufigkeit auf die unterschiedlichen Daten aber unterschiedlich sein kann, ist man an der Minimierung der mittleren Pfadlänge l_m des Baumes interessiert. Analog stellt sich bei der Nachrichtenübertragung das Problem die mittlere Codewortlänge l_m zu minimieren, wobei die unterschiedliche Auftrittswahrscheinlichkeit der Quellenbuchstaben berücksichtigt werden sollte. Sei (Y, P) eine Quelle, $p : Y \rightarrow [0,1]$ mit $\sum_{y \in Y} p(y) = 1$, die mittlere Codewortlänge bzw. mittlere Pfadlänge errechnet sich aus $l_m = \sum_{y \in Y} p(y) n_y$ mit n_y Länge des Pfades von der Wurzel des Baumes zum Knoten y, bzw. $k_y = |\varphi(y)|$. Im folgenden betrachten wir nur noch das Problem der Minimierung der mittleren Codewortlänge, das zweite Problem erschießt man durch Analogiebetrachtungen. Ein erstes Prinzip bei der Konstruktion des Codes mit minimaler Codewortlänge wird sein : $p(y) \leq p(y') \rightarrow |\varphi(y)| \geq |\varphi(y')|$, was sonst.

Definition 2.4 (Information und Entropie) Sei (Y, P) eine gegebene Quelle, so setze $I(y) = -\log(p(y))$ und bezeichne den Wert $I(y)$ als Information des Buchstaben y. Die Funktion I ist ein Gradmesser für die Ungewißheit die dem Buchstaben y 'anhaftet'. Da I Zufallsgröße ist, kann der Erwartungswert von I gebildet werden, setze $H_P = E(I) = -\sum_{y \in Y} p(y) log(p(y))$. Der Wert H_P wird als Entropie bezeichnet, er spielt nicht nur in der Informationstheorie eine Rolle, sondern findet auch in der Physik (Thermodynamik). sein Anwendung.

Eine weitere offensichtliche Frage ist die Frage nach einer unteren Abschätzung der mittleren Codewortlänge, diese wird im folgenden Satz beantwortet.

Satz 2.7 (Abschätzung für minimale mittlere Codewortlänge) *Es sei (X, P) eine Quelle, $\varphi : Y \to X$ eine eindeutige Codierung, so gilt $l_m \geq H_P$.*

Beweis. Es gilt die Abschätzung $\ln(a) \leq a - 1 \forall a > 0$ bzw. $a \leq \exp(a-1) \forall a > 0$. Dies sieht man z.B. folgendermaßen : $a-1 < 0 < exp(a-1) \forall a \in [0, 1)$, bzw $1-1 = 0 < \exp(1)$ und $(a-1)' = 1 \leq (\exp(a-1))' = \exp(a-1) \quad \forall a \geq 1$. $H_P - l_m = -\sum y \in Y p(y)(|\varphi(y)| + \log_r(p(y)) = \frac{1}{\ln(r)} \sum_{y \in Y} (r^{-|\varphi(y)|} - p(y)) \leq 1$, dies folgt aus mehr oder weniger bekannten Logarithmengesetzen und der Ungleichung von Kraft, die für einen eindeutigen Code gültig ist, d.h $H_P \leq l_m$. □

Satz 2.8 (Shannon, ca. 1945) *Zu jeder Häufigkeitsverteilung $PP = (p_1, \ldots, p_n)$ gibt es einen Präfix-Code $C \subseteq X^*, C = \{w_1, \ldots, w_n\}$, mit $|X| = r$, X vorgegeben, mit einer mittleren Codewortlänge l_m, die der Abschätzung $l_m \leq H_P + 1$ genügt.*

Beweis. Sei o.B.d.A. $0 < p_i \leq p_j (i \leq j)$, $p_i > 0$ wird nur aus Übersichtlichkeitsgründen vorausgesetzt, die Wahrscheinlichkeitsverteilung P muß insbesondere auch nicht endlich sein. Wir setzen $l_i = \lceil -\log(p_i) \rceil \leq -\log(p_i) < -\log(p_i) + 1$, mit diesen Längen würde sich l_m zu $l_m = \sum_{i=1}^{n} p_i * l_i < \sum i = 1^n - p_i * (\log(p_i) + 1) = H_P + 1$ ergeben, d.h. falls ein Präfix-Code $C = \{w_1, \ldots, w_n\}$ mit $|w_i = l_i$ existiert, so erfüllt dieser die gewünschte Abschätzung. Setze $k_0 = 0, k_i = k_{i-1} + p_i$, betrachte die Darstellung von k_i im Zahlensystem zur Basis r. (wähle falls möglich die abbrechende Darstellung, d.h. nicht $0.00111\ldots$ sondern $0.0100\ldots$ für $k_i = 0.25$). Setze w_i gleich den ersten l_i Stellen nach dem Komma von K_i's Entwicklung. Es gilt $C = \{w_i\}$ ist ein Präfix-Code. Zu zeigen ist nur $w_i \not\sqsubseteq w_j (i < j)$, da $l_i \leq l_j$. Angenommen $w_i \sqsubseteq w_j (i < j)$, so gilt per Definition $k_i = 0.w_i \ldots$, damit ist $w_i \sqsubset w_{j-1} \sqsubset k_{j-1} \to |k_{j-1} - k_{i-1}| < \frac{1}{r^{|w_i|}} = r^{l_i} \leq p_i$, aber $|r_{i-1} - r_{j-1}| \geq p_i$ nach Konstruktion, Widerspruch. □

Beispiel 2.3 $P = (p_1 = 1/2, p_2 = 1/4, p_3 = 1/8, p_4 = 1/14, p_5)$, p_5 Restbetrag. $\to l_1 = 1, l_2 = 2, l_3 = 3 \; l_4 = 4, l_5 = 5, p_1 = 0.1, p_2 = 0.01, p_3 = 0.001, p_4 = 0.0001 \cdots \to w_1 = 0, w_2 = 01 \; w_3 = 110, w_4 = 1110, w_5 = 11110$.

Die im Beweis aufgetretene Codierungsvorschrift liefert den Shannon-Code. Der Beispiel Code ist nicht optimal, im weiteren wenden wir uns nun der Frage der Konstruktion eines optimalen Codes zu. Zunächst gibt es optimale Codes C, da diese gewisse l_i festlegen, die der Ungleichung von Kraft genügen, folgt auch die Existenz eines Präfix-Codes C' mit $l'_i = l_i$. Sei $C = \{w_1, \ldots, w_n\}$ optimaler Code und $p_1 \geq \cdots \geq p_n$ so folgt sofort $|w_1| \leq \cdots \leq |w_n|$. Es gilt jedoch auch das folgende Lemma:

Lemma 2.9 Sei $C = \{w_1, \ldots, w_n\}$ optimaler Präfix-Code (Codewörter geordnet in quasilexikographische. Ordnungsrelation), mit $p_1 \geq \cdots \geq p_n$, so gilt $|w_{n-1}| = |w_n|$ und w_{n-1}, w_n unterscheiden sich nur im letzten Buchstaben.

Beweis. Angenommen $|w_{n-1}| < |w_n| \to w_n = w'_n aa \in X$ und mit $w_i \not\sqsubseteq w_n$ folgt $w_i \not\sqsubseteq w'_n$ für $(i < n)$, ersetze w_n durch w'_n und erhalte dadurch Code mit geringerer mittlerer Wortlänge, beachte $|w_n| > |w_i| \quad (i < n)$, d.h. $w'_n \not\sqsubseteq w_i$ aus Gründen der Wortlänge, Widerspruch zur Minimalität von C. Angenommen w_n und w_{n-1} unterscheiden sich eher, so ist $w'_n \not\sqsubseteq w_{n-1}$ nach Annahme. Sei nun $|w_i| = |w_{n-1}| = |w_n|, w_i < w_{n-1} < w_n$, so

gilt $w_i' \leq w_{n-1}' < w_n' \rightarrow w_i' \not| w_n'$, d.h. $w_n' \not\sqsubset w_i$. Sei nun $|w_i| < |w_n|$ und angenommen $w_n' \sqsubset w_i \rightarrow |w_i| = n-1$ und somit $w_n' = w_i \sqsubset w_n$, Widerspruch zu präfix Eigenschaft des Codes C, d.h. Ersetzung von w_n durch w_n' liefert Code mit geringerer mittlerer Wortlänge, Widerspruch zur Minimalität. □

Lemma 2.10 (Huffman - Codierung, binäres Alphabet) Es sei $X = \{0,1\}$,$P = (p_1, ;p_n), p_i \geq p_j (i \leq j), p_k > 0, p_{n-1,n} = p_{n-1} + p_n$, C' optimaler Präfix-Code für $P' = (p_1, \ldots, p_{n-2}; p_{n-1,n})$, so ist $C = C' \setminus \{w_{n-1}\} \cup \{w_{n-1}0, w_{n-1}1\}$ optimaler Präfix-Code für P.

Beweis. Es ist nur Schreibarbeit zu zeigen C Präfix-Code, gezeigt wird hier deshalb nur die Optimalität von C. Es gilt $l_{C,P} = l_{C',P'} + p_{n-1} + p_n$, mit $l_{C,P}, l_{C',P'}$ mittlere Codewortlängen bzgl. der jeweiligen Wahrscheinlichkeitsverteilung. Sei $C'' = \{w_1'', \ldots, w_n''\}$ optimaler Code für P, da w_{n-1}'', w_n'' nur an letzter Stelle verschieden setze $C''' = \{w_1'', \ldots, w_{n-1}'''\}$ mit $w_{n-1}''' = w_{n-1}aa \in X \rightarrow C'''$ Präfix-Code und $l_{C'',P'} = l_{C''',P'} + p_{n-1} + p_n \geq l_{C',P'} + p_{n-1} + p_n = l_{C,P}$ $\rightarrow C$ optimaler Code. □

Bemerkung 2.2 Der Huffman-Algorithmus ergibt sich nun zum Beispiel folgendermaßen

- sortiere Wahrscheinlichkeiten von P in fallender Ordnung $\rightarrow P = (p_1, \ldots, p_n)$
- setze $P' = (p_1, \ldots, p_{n-2}, p_{n-1,n})$ und konstruiere Code für $(n-1)$ Häufigkeiten $\rightarrow$ Lemma gibt nun Code für n Häufigkeiten
- $n = 2 \rightarrow C = 0, 1$

Zur weiteren Beschäftigung mit Huffman Codes siehe z.B. [4] , dort wird gezeigt, daß die Konstruktion des Huffman Codes mittels eines Greedy-Algorithmus durchgeführt werden kann.

Die binäre Huffman-Codierung ergibt einen Suchbaum mit n Blättern mit der Eigenschaft, daß jeder innere Knoten genau zwei Nachfolger besitzt. Man sieht schnell, daß es für ein allgemeines Paar n (Anzahl der Blätter), r (Verzweigungsgrad der inneren Knoten) keinen Baum mit der Eigenschaft, daß jeder innere Knoten den Verzweigungsgrad r besitzt, geben muß. Die Konsequenz dieser Tatsache spiegelt sich in einer leichten Modifikation des Huffman-Algorithmus wieder.

Lemma 2.11 (Huffman - Codierung für r-näres Alphabet) Es sei X gegebenes Alphabet, $|X| = r, n = 1 + k*(r-1) + r'r' < r$ und $p1 \geq \cdots \geq p_n$. Ist C' ein optimaler Präfix-Code zur folgenden Verteilung $P = \{p_1, \ldots, p_{1+n*(r-1)}, p'\}$ mit $p' = p_{1+n*(r-1)+1} + \cdots + p_n$ mit den Codewörtern $w_1, \ldots, w_{1+k*(r-1)}, w$,so ist $C := (C' \setminus \{w\}) \cup \{wa_1, \ldots, wa_{r'+1}\}$ optimaler Präfix-Code zur Ausgangsverteilung.

Bemerkung 2.3 Der allgemeine Huffman-Algorithmus ergibt sich nun zum Beispiel folgendermaßen

- sortiere Wahrscheinlichkeiten von P in fallender Ordnung $\rightarrow P = (p_1, \ldots, p_n)$
- setze $P' = (p_1, \ldots, p_{1+k*r}, q) q = p_{1+n*r+1} + \cdots + p_n$ und konstruiere Code C' für $(n - r')$ Häufigkeiten,
  setze $r' = r - 1$ falls $n = 1 + k*(r-1)$ oder $r' = (n-1) \mod (r-1)$ falls $n = 1 + k*(r-1) + s \quad 0 < s < r-1$).

- $C = (C' \setminus \{w\}) \cup \{wa_1, \ldots, wa_{r'}\}$

Im vorhergehenden Abschnitt wurde sich mit der Minimierung der mittleren Codewortlänge beschäftigt, es wurde weiterhin bereits darauf hingewiesen daß diese Aufgabe in einem engen Zusammenhang zur Minimierungsaufgabe in einem Suchbaum steht. Es wäre jedoch wünschenswert die ursprüngliche Ordnung beizubehalten, d.h $a_i \leq a_j \rightarrow \varphi(a_i) \leq \varphi(a_j)$, φ Codierungsvorschrift. Eine solche Codierung wird alphabetischer Code genannt.

Satz 2.12 (Existenz und Konstruktion alphabetischer Codes, Gilbert, Moore) *Es seien $a_i \quad i = 1, \ldots, n$ Quellensymbole und p_i zugeordnete Auftrittswahrscheinlichkeiten. Dann gibt es eine alphabetische Codierung $\varphi : Y^* \rightarrow X^*$, derart daß die mittlere Codewortlänge der Abschätzung $l_m \leq H_P + 2$ genügt.*

Beweis. Analog zur Shannon Codierung setze $r_0 = \frac{p_1}{r}, r_i = \sum_{k=1}^{i-1} p_k + \frac{p_i}{r}, l_i = \lceil \log(\frac{1}{p_i}) \rceil + 1$, entwickle r_i r-adisch und setze $w_i \sqsubseteq r_i$ mit $|w_i| = l_i$. Der so konstruierte Code ist präfixfrei. Da $r_i \leq r_j$ für $i \leq j$ folgt $w_i \leq w_j$ in lexikographischer Ordnungsrelation, noch zu zeigen bleibt $w_j \not\sqsubseteq w_i$ für $i < j$. Angenommen es wäre so, so folgt wie im Beweis zur Shannon Codierung $|r_i - r_j| < \frac{1}{r^{|w_i|}} \leq \frac{p_i}{r}$, andererseits $|r_i - r_j| \geq \frac{r-1}{r} * p_i + \frac{p_j}{r} \geq \frac{p_i}{r}$, Widerspruch. Den Beweis der Ungleichung überlasse ich dem interessierten Leser dieser Seiten. (Siehe Beweis zur Shannon Codierung). □

Bemerkung 2.4 Wir wissen $l_m \geq H_P$ für gegebene Quelle (Y, P), angenommen wir betrachten die Quelle (Y^2, P^2) und es gilt $p(a_i a_j) = p(a_i) * p(a_j) \; \forall a_i, a_j \in Y$, d.h das Auftreten der Buchstaben ist unabhängig voneinander, so kann man mittels leichter algebraischer Umformungen zeigen, daß $H_{P^n} = n * H_P$ gilt. Mittel dieser Tatsache läßt sich nun der folgende Satz beweisen.

Satz 2.13 (Shannons Quellenkodierungssatz) *Zu jeder Quelle (Y, P), die die Bedingung $p(a_i a_j) = p(a_i) * p(a_j) \forall a_i, a_j \in Y$ erfüllt, und jedem $\epsilon > 0$ gibt es $n \in \mathbb{N}$ und eine Codierung $\varphi : Y^n \rightarrow X^*$ derart daß die mittlere Codewortlänge für einen Quellenbuchstaben der Ungleichung $l_{p^n,1} \leq H_P + \epsilon$ genügt.*

Beweis. Es gibt Codierung φ für die $l_{\varphi,P^n} \leq H_{P^n} + 1 = n * H_P + 1$, (Shannon Codierung), d.h. $l_{p^n,1} = \frac{1}{n} * l_{\varphi,P^n} \leq H_p + \frac{1}{n}$. Wähle n hinreichend groß und die Abschätzung wird erfüllt. □

Bemerkung 2.5 Ist das Auftreten der Buchstaben nicht unabhängig voneinander, so gilt allgemein $H_{P^n} \leq n * H_P$, d.h die Codierung in Blöcken würde in diesem Falle eine noch kleinere mittlere Codewortlänge versprechen.

Bemerkung 2.6 (Ein Adaptives Codierungsverfahren) Jedes Verfahren besteht aus den Stufen Schätzen der Häufigkeit, Codierung nach Schätzung, Anpassung an aktuelle Schätzung. Wir betrachten im folgenden eines der einfachsten Verfahren, genannt Move-to-Front Codierung.

- Start mit einer Ausgangscodierung $C = \{w_1, \ldots, w_n\}, |w_i| \leq w_j | i \leq j$ des Alphabets $Y = \{a_1, \ldots, a_n\}$ und der Codiervorschrift $\varphi_0(a_i) = w_i$, wobei Codewort und das zu codierende Symbol sich in ein und derselben Spalte einer Codierungstabelle befinden.

- Gegeben sei Nachricht $y_1 \dots y_n$, Zum Zeitpunkt j Verwendung der Codierungsvorschrift $\varphi_{j-1}(a_i) = w_{k_i}$, wobei w_{k_i} das zum Zeitpunkt $j-1$ dem Symbol a_i zugeordnete Codewort ist.

- Anpassungsschritt: Weise dem Symbol a_i das Codewort w_1 (Move-to-Front) und lasse die Symbole in der Codierungstabelle nach unten rutschen, am besten siehe Beispiel.

- Die Decodierung läuft analog, da Ausgangscodierung bekannt, d.h. das erste Codewort ist somit decodierbar und die Codierungstabelle des 1. Schrittes kann somit rekonstruiert werden.

Es ist offensichtlich das die mittlere Codewortlänge für Symbole mit relativ hoher Auftrittswahrscheinlichkeit relativ gering gering ist.

Beispiel 2.4

- Ausgangstabelle, Codierungstabelle zum Zeitpunkt 0

  - $w_1 \to a_1$
  - $w_2 \to a_2$
  - $w_3 \to a_3$

Die zu codierende Nachricht sei $a_3a_1a_2a_2 \dots$

- Codierungsstabelle zum Zeitpunkt 1

  - $w_1 \to a_3$
  - $w_2 \to a_1$
  - $w_3 \to a_2$

- Ausgangstabelle,Codierungstabelle zum Zeitpunkt 2

  - $w_1 \to a_1$
  - $w_2 \to a_3$
  - $w_3 \to a_2$

- Ausgangstabelle,Codierungstabelle zum Zeitpunkt 3 und zum Zeitpunkt 4

  - $w_1 \to a_2$
  - $w_2 \to a_1$
  - $w_3 \to a_3$

- $\dots$

Die Codierte Nachricht hat die Form $w_3w_2w_3w_1 \dots$.

3 Sprachentheoretische Eigenschaften von Codes

Definition 3.1 (Maximale Codes) $C \subseteq X^*$ heißt maximaler Code $\iff \forall W \subseteq X^*$ mit den Eigenschaften W Code und C ist eine Teilmenge von W folgt $W = C$, d.h. es gibt kein Wort in $X^* \setminus C$ welches man zu C hinzufügen kann und die entstehende Teilmenge bleibt immer noch ein (eindeutig decodierbarer) Code.

Satz 3.1 (Über die Existenz von maximalen Codes) *Zu jeden Code $C \in X^*$ gibt es einen maximalen Code der C enthält.*

Beweis. Definiere induktiv die folgende Familie von Teilmengen aus X^*. $C_0 = C, C_{n+1} = C_n \cup W_n$ mit $W_n \subseteq X^n \setminus C_n$ und W_n maximal, so daß C_{n+1} ein Code bleibt, d.h. W_n besteht aus allen Wörtern der Länge $|n|$, die man zu C_n hinzufügen kann ohne die Codeeigenschaft zu verletzen. (Beachte insbesondere W_n ist eine endliche Menge.) Setze $\mathbb{C} = \cup_{n=1}^{\infty} C_n$. Es gilt nun

- alle C_n sind Codes und $C_i \subseteq C_j \quad i \leq j$, per Konstruktion
- $\mathbb{C}$ ist Code
- $\mathbb{C}$ ist maximal

Seien nun $w_1, \ldots, w_k, v_1, \ldots, v_k \in \mathbb{C}$ mit der Eigenschaft $w_1 \ldots w_k = v_1 \ldots v_r$ gegeben, so existiert $n \in \mathbb{N}$ mit $w_i, v_j \in C_n$, da C_n Code folgt jedoch $k = r, w_i = v_i$, d.h. die Codeeigenschaft ist auch für $\mathbb{C}$ erfüllt. (z.B. definiere n durch $n = 1 + max\{|w_i|, |v_j|, 1 \leq i \leq k, 1 \leq j \leq r\}$) Angenommen $\mathbb{C}$ ist nicht maximal, d.h. es gibt $w \in X^*$ welches $\mathbb{C}$ hinzugefügt werden kann, aber mit $|w| = n$ steht die Existenz eines solchen Wortes w im Widerspruch zur Konstruktion von C_{n+1}, da $C_{n+1} \cup \{w\}$ auch ein Code wäre, aber $C_{n+1} = C_n \cup W_n$ mit W_n maximal. □

Bemerkung 3.1 Ein Nachtrag zum letzten Satz. Markov zeigte 1967, daß nicht zu jedem endlichen Code ein endlicher maximaler Code existieren muß. Insbesondere zeigte er das jeder maximale Code der $C = \{0, 10, 011, 11111\}$ enthält unendlich sein muß. Dieses Ergebnis ist nicht trivial und war bis in die 70er Jahre relativ unbekannt. Der Beweis der Behauptung wird im Abschnitt Maximalität und Vollständigkeit geführt werden.

3.1 Der Satz von Sardinas und Petterson

Definition 3.2 ($C - Kette$) Sei $w, v \in X^*$ und $C \subseteq X^*$, so definiere die folgenden zwei Relationen

- $w <_1 v \iff \exists u \in C$ mit $u = wv$, d.h. $wv \in C$
- $w <_2 v \iff \exists u \in C$ mit $w = uv$

Eine Folge von Wörtern $\{v_i\}$ bildet eine $C-Kette$ für bzgl. einer Teilmenge C, wenn $v_1 \in C, v_1 \neq v_2$ und $v_1 <_{i_1} v_2 <_{i_2} \cdots <_{i_{n-1}} v_n$ mit $i_k \in \{1, 2\}$. Die Relationen $<_1, <_2$ wurden von Levenshtejn eingeführt, und der hier vorgestellte Beweis des Satzes von Sardinas und Patterson geht auf ihn zurück.

Lemma 3.2 (Levenshtejn) Eine Folge von Wörtern $\{v_i\}_1^n$ bildet eine $C - Kette$ der Länge $n \iff$ es gibt Wörter $w_1, \ldots, w_i \in C$ und $u_1, \ldots, u_j$ in C mit $w_1 \neq v_1$ und $i+j = n$ derart, daß $w_1 \ldots w_{i-1} \sqsubseteq u_1 \ldots u_j \sqsubseteq w_1 \ldots w_i$ $(*)$ gilt, es gilt dann auch weiterhin $w_1 \ldots w_i = u_1 \ldots u_j v_n (**)$. Die Forderung $(*)$ soll garantieren, daß $<_1, <_2$ alternierend erscheinen.

Beweis. Die Hinrichtung erfolgt durch Induktion über n, d.h. Länge der Kette

(IA) Für $n = 2$ gilt mit $v_1 <_1 v_2 \to v_1 v_2 \in C$ also $e \sqsubseteq v_1 \sqsubseteq v_1 v_2$, bzw. mit $v_1 <_2 v_2 \to \exists u \in C$ mit $v_1 = u v_2$, d.h. $e \sqsubseteq u \sqsubseteq v_1$ mit $i = j = 1$ in beiden Fällen.

(IV) Lemma gelte nun für $n \geq 2$.

(IB) Aus der Gültigkeit des Lemmas für n folgt Gültigkeit des Lemmas für $n + 1$.

(IS) Nach (IV) existieren $w_i \in C, u_j \in C$ mit $w_1 \ldots w_{i-1} \sqsubseteq u_1 \ldots u_j \sqsubseteq w_1 \ldots w_i$ und $w_1 \ldots w_i = u_1 \ldots u_j v_n$. Angenommen $v_n <_1 v_{n+1} \to v_n v_{n+1} \in C$ und damit $w_1 \ldots u_j \sqsubseteq w_1 \ldots w_i \sqsubseteq u_1 \ldots u_j u_{j+1}$ mit $u_{j+1} = v_n v_{n+1}$, andererseits für $v_n <_2 v_{n+1} \exists u$ mit $v_n = u v_{n+1} \to w_1 \ldots w_{i-1} \sqsubseteq u_1 \ldots u_j u \sqsubseteq w_1 \ldots w_i$ und $u_1 \ldots u_j u v_{n+1} = u_1 \ldots u_j v_n = w_1 \ldots w_i$.

Die Rückrichtung erfolgt ebenfalls über die Länge der C-Ketten.

(IA) $n = 2$ Wir haben $e \sqsubseteq u_1 \sqsubseteq w_1, w_1 \neq v_1$, d.h. es existiert v_2 mit $u_1 v_2 = w_1 v_2 \neq e$

□

Folgerung 3.3 *$C \subseteq X^*$ ist Code $\iff$ es gibt keine endliche C-Kette, die das leere Wort enthält. Dies folgt mehr oder weniger direkt aus $(**)$ und der Beziehung $n = i + j$.*

Eine andere Formulierung dieser Folgerung ist der Satz von Sardinas und Patterson.

Satz 3.4 (Satz von Sardinas und Patterson) *Es sei $C \subseteq X^* \setminus \{e\}$. Wir definieren induktiv die folgende Familie von Teilmengen aus X^*.*

- $U_1 = C$
- $U_2 = \{v | v \neq e, \exists u \in C u <_2 v\}$
- $U_i = \{v | \exists u \in U_{i-1} mit\ u <_k vk \in \{1, 2\}\}$

Dann gilt C ist Code $\iff \forall i \in \mathbb{N} e \notin U_i$.

Beweis. Es genügt zu bemerken, daß die U_i Enden entsprechender C-Ketten sind. □

Bemerkung 3.2 Für die nachfolgenden Lemmata führen wir eine Notation ein, wir setzen für $L, W \subseteq X^*$

- $L^{-1}W = \{v \in X^* | \exists u \in L, v \in W \text{ mit } uv = w\}$
- $WL^{-1} = \{v \in X^* | \exists u \in L, v \in W \text{ mit } vu = w\}$

Mit dieser Notation, der Notation im Satz von Sardinas und Patterson und der Definition für $<_2$ folgt $U_2 = C^{-1}C \setminus \{e\}, U_i = U_{i-1} \cup C^{-1}U_{i-1}$. Dies gilt, da $U_i = \{v|\exists w \in U_{i-1}$ mit $w <_1 v\} \cup \{v|\exists w \in U_{i-1}$ mit $w <_2 v\}$. Benutze Definition von $<_1$ bzw. $<_2$ und man sieht sofort, daß es sich nur um eine andere Schreibweise handelt.

Lemma 3.5 Es sei $\sim$ eine Kongruenzrelation auf X^*, und es sei $L \subseteq X^*$ eine Vereinigung von Kongruenzklassen von $\sim$, d.h. $L = \{w|\exists v \in L$ mit $w \sim v\}$. Dann ist für beliebiges $U \subseteq X^*$ auch $U^{-1}L$ eine Vereinigung von Kongruenzklassen von $\sim$.

Beweis. $v \in U^{-1}L, v \sim w \rightarrow \exists u \in U$ mit $uw \in L$, da $\sim$ Kongruenzrelation folgt $uw \sim uv$ und nach Voraussetzung über L folgt somit für ein $l' \in Luv = l' \rightarrow v \in U^{-1}L$. $\square$

Lemma 3.6 Ist $C \subseteq X^* \setminus \{e\}$ regulär, so gibt es eine Kongruenzrelation von endlichem Index derart, daß die U_i aus dem Satz von Sardinas und Patterson darstellbar sind als Vereinigung von Kongruenzklassen von $\sim$, insbesondere sind alle U_i regulär und die Anzahl verschiedener U_i ist endlich.

Beweis. Wir wissen C reguläre Sprache $\iff$ die syntaktische Kongruenzrelation $\sim_C$ ist von endlichem Index. Da C regulär und $U_1 = C$ folgt für U_1 die Behauptung sofort bei Verwendung von $\sim_C$. Für $U_2 = C^{-1}C \setminus \{e\}$ definiere eine etwas andere Kongruenz, definiere $\sim_{C'}$ durch $w \sim_{C'} v \iff w \sim_C v, w, v \neq e$ oder $v = w = e$. Diese Definition ergibt wiederum eine Kongruenzrelation (wäre zu beweisen) deren Index der Abschätzung $Index \sim_{C'} \leq Index \sim_C +1$ genügt. Zu zeigen U_i sind Vereinigung von Kongruenzklassen von $\sim_{c'}$. Dies erfolgt durch Induktion über i.

(IA) $i = 1$ Satz von Kleene, Myhill und Nerode, $i = 2$ Verwendung des letzten Lemmas unter Berücksichtigung des Ausschlusses von e.

(IV) Lemma gelte nun für $n \geq 2$.

(IB) Aus der Gültigkeit des Lemmas für n folgt Gültigkeit des Lemmas für $n + 1$.

(IV) $U_{n+1} = U_n^{-1}C \cup C^{-1}U_n$, C ist eine Vereinigung von Kongruenzklassen nach dem Satz von Kleene ..., U_n ist es nach Induktionsvoraussetzung, bei Verwendung des letzten Lemmas folgt somit die Gültigkeit der Induktionsbehauptung für $n + 1$.

Um U_i regulär zu zeigen, reicht es aus zu zeigen, daß die Kongruenzklassen regulär sind. (Wir wissen bereits das die Anzahl der Kongruenzklassen endlich ist, und es gilt ebenfalls daß die Vereinigung endlich vieler regulärer Mengen wiederum etwas reguläres ist.) $\square$

Folgerung 3.7 *Ist $C \subseteq X^*$ regulär, so ist die Eigenschaft C ist Code, entscheidbar in endlicher Zeit. Dieses Problem ist schon für kontextfreie Sprachen nicht mehr entscheidbar in endlicher Zeit.*

Bemerkung 3.3 Der obige Satz führt auf einen Algorithmus zur Entscheidung der Frage ist C ein Code, C natürlich regulär. Bestimme die U_i (nur endlich viele) in Form von regulären Ausdrücken und entscheide ob leeres Wort in einem der U_i enthalten ist. Dieses Verfahren ist natürlich etwas mühsam und auch nur schlecht programmtechnisch zu realisieren. Eine andere Methode soll im folgenden vorgestellt werden. Da C reguläre Sprache existiert ein NDA $\mathbb{A}$ (nichtdeterministischer endlicher Automat) der die Sprache entscheidet.

Automatenkonstruktion für Test auf Code-Eigenschaft

gegeben: NEA $\mathcal{A} = (Z, X, I, \delta, F)$
Frage: Ist $L = L(\mathcal{A})$ ein Code?

1. Konstruiere $\mathcal{A}^* = (Z^*, X, I^*, \delta^*, F^*)$ mit $L(\mathcal{A}^*) = L^*$, wobei o.B.d.A. $Z^* \cap Z = \emptyset$.
2. Konstruiere $\mathcal{A}' = (Z', X, I', \delta', F')$, wobei
   (a) $Z' = (Z \cup Z^*)^2,\ I' = I \times I,\ F' = F^* \times F^*$,
   (b) $\delta' = \delta_0 \cup \delta_1 \cup \delta_2$ mit

$$\begin{aligned}
\delta_0 &= \{((y_1, y_2), x, (z_1, z_2)) : (y_i, x, z_i) \in \delta \cup \delta^*, i = 1, 2\},\\
\delta_1 &= \{((y_1, y_2), x, (z_1, z_2)) : (y_2, x, z_2) \in \delta \cup \delta^* \wedge z_1 \in I^* \wedge \exists q_1(q_1 \in F \wedge (y_1, x, q_1) \in \delta)\},\\
\delta_2 &= \{((y_1, y_2), x, (z_1, z_2)) : (y_1, x, z_1) \in \delta \cup \delta^* \wedge z_2 \in I^* \wedge \exists q_2(q_2 \in F \wedge (y_2, x, q_2) \in \delta)\}.
\end{aligned}$$

Beide Komponenten simulieren zunächst (durch δ_0) einen Lauf von $\mathcal{A}$. Wird nach Eingabe von v_i in Komponente $i \in \{1, 2\}$ ein Zustand aus F erreicht, so darf (nach δ_i) in dieser Komponente mit der Simulation von $\mathcal{A}^*$ (nach δ_0) begonnen werden. Durch die Wahl von δ_1 und δ_2 ist garantiert, daß $v_1 \neq v_2$ gilt. Es gilt $L(\mathcal{A}') = \{w \mid \exists v_1, v_2 \in L, w_1, w_2 \in L^* : w = v_1 w_1 = v_2 w_2 \wedge v_1 \neq v_2\}$, d.h., $L(\mathcal{A})$ ist genau dann ein Code, wenn $L(\mathcal{A}')$ leer ist. Dies ist effektiv entscheidbar

3.2 Maximalität und Vollständigkeit

Im folgenden versuchen wir die Eigenschaft maximaler Code zu sein näher zu charakterisieren, da man der Auffassung sein könnte, daß die Eigenschaft 'maximal' für einen Code etwas positives ist, z.B. würde Maximalität heißen, das vorgegebene Alphabet effizient ausgenutzt zu haben. Eine hinreichende Bedingung für die Maximalität eines Codes liefert der Satz von MacMillan : Wenn ein Bernoulli-Maß $\mu : X^* \to [0, 1]$ existiert mit $\mu(C) = 1$ folgt C ist maximaler Code.

Definition 3.3 (Dichte und Magere Sprachen) Eine Sprache $L \subseteq X^*$ heißt dicht $\iff \forall w \in X^*$ gilt $L \cap X^* w X^* \neq \emptyset$. Bezeichne mit $T(L)$ die Menge $T(L) = \{v : \exists u, w \in X^*$ mit $uvw \in L\}$ der Worte die als Infix in einem Wort aus L auftreten, so gilt L dicht $\iff T(L) = X^*$. Eine Sprache heißt mager, falls sie nicht dicht ist.

Beispiel 3.1
- $X = \{a, b\}\ L = \cup_{n=1}^{\infty} a^n b X^n$, L ist dicht, wie man hoffentlich leicht sieht.
- $X + \{a, b\}L' = \{bb, babb, bbab\}^*$, L' ist nicht dicht, da $aba \notin T(L')$

Definition 3.4 (Vollständigkeit eines Erzeugendensystems) Eine Teilmenge $L \subseteq X^*$ heißt vollständig $\iff L^*$ ist dicht.

Folgerung 3.8
- *Ist $C \subseteq X^*$ ein Code, $\mu : X^* \to [0, 1]$ ein Bernoulli-Maß, so gilt* $\mu(C^*) = \infty \iff \mu(C) = 1$

- *Ist* $L \subseteq X^*$ *und* $\mu(L^*) = \infty \rightarrow \mu(L) \geq 1$

Beweis. Der Beweis erfolgt mittels Anwendung der beim Beweis des Satzes von MacMillan hergeleiteten Beziehungen $\mu(M^i) \leq \mu(M)^i$, wobei das $=$ steht falls M ein Code. Und da $L^* = \cup_{n=0}^{\infty} L^i$ folgt aufgrund der Additivitätseigenschaft des Bernoulli-Maßes $\mu(L^*) \leq \sum_{n=0}^{\infty} \mu(L)^n$ bzw. $\mu(C^*) = \sum_{n=0}^{\infty} \mu(C)^n$. Der Rest folgt bei Verwendung der jeweiligen Voraussetzung sowie der Summenformel für geometrische Reihen. □

Lemma 3.9 Ist $\mu : X^* \rightarrow [0,1]$ ein Bernoulli - Maß und ist L magere Sprache, so gilt $\mu(L) < \infty$.

Beweis. L magere Sprache $\rightarrow \exists w \notin T(L) \rightarrow L \subseteq M = X^{\leq |w|}(X^{|w|} \setminus \{w\})^*$, es gilt somit $\mu(L) \leq \mu(M)$, d.h. $\mu(L) \leq C * \sum_{n=0}^{\infty} q^i$ mit $C = \mu(X^{|w|}), q = \mu(X^{|w|} \setminus \{w\})$, und da $q < 1$ folgt die Behauptung. Im letzten Schritt wurde ebenfalls die in der letzten Folgerung verwendete Abschätzung benutzt. □

Bemerkung 3.4 Es gibt dichte Sprachen mit $\mu(L) < \infty$, Beispiele hierfür findet man am Ende dieses Kapitels.

Folgerung 3.10 *Es sei* $\mu : X^* \rightarrow (0,1]$ *ein Bernoulli - Maß. Ist* L *magere Sprache und vollständig, so gilt* $\mu(L^*) = \infty$.

Beweis. $w \notin T(L)$, L^* dicht $\rightarrow \forall v \in X^* wvw \in T(L^*)$. Betrachtet man eine Überdeckung von wvw durch Worte u_j aus L, so sieht man sofort das folgende : Es gibt $w_1, w_2 \in X^*$ mit $w_1 v w_2 \in L^*, |w_1|, |w_2| < |w|$. Natürlich sind die w_i abhängig"von v. Also existiert eine Abbildung $f : X^* \rightarrow L^*$ mit $v \rightarrow w_1 v w_2$, wobei man für jedes v ein Paar w_1, w_2 nach Lust und Laune bestimmen kann, auf jeden Fall jedoch eindeutig. Man hat die folgenden Eigenschaften

- $|f(v)| \leq |v| + 2 * (|w| - 1)$
- $|f^{-1}(u)| \leq 2 * |w|$, wobei $f^{-1}(u)$ die Menge aller Urbilder ist, d.h. $f(x) = u \rightarrow x \in f^{-1}(u)$
- $\mu(v) \geq \mu(f(v))$

Setze $\mu_{min} = min\{\mu(x) : x \in X\} \rightarrow 1 > \mu(u) \geq \mu_{min}^{|u|} > 0$ (siehe u.a. Voraussetzung an das Bernoulli - Maß). Setze weiterhin $c = \mu_{min}^{|w|} \rightarrow \mu(v) * c^2 \leq \mu(v)$. Mit den Eigenschaften von f folgt $\mu(f^{-1}(u)) \leq \frac{2*|w|}{c^2} * \mu(v)$. Da L vollständig folgt $f^{-1}(L^*) = X^*$ und somit die folgende Ungleichungskette : $\mu(X^*) = \mu(f^{-1}(L^*)) \leq \sum_{u \in L^*} \mu(f^{-1}(u)) \leq \frac{2*|w|}{c^2} * \mu(L)$, d.h. $\mu(X^*) \leq C' * \mu(L^*), C' \neq 0 \rightarrow \mu(L^*) = \infty$, d.h. die Behauptung. □

Folgerung 3.11 *Ist* $C \subseteq X^*$ *ein magerer vollständiger Code, so gilt für alle Bernoulli-Maße* $\mu : X^* \rightarrow (0,1]$ *die Beziehung* $\mu(C) = 1$

Satz 3.12 (Vervollständigungssatz von Ehrenfeucht und Rozenberg) *Es sei* $C \subseteq X^*$ *ein Code und* y *ein unberandetes Wort mit* $y \notin T(C^*)$. *Weiter sei* $U = (X^* \setminus C^*) \setminus X^* y X^*$, *dann ist* $C' = C \cup U$ *ein vollständiger Code der* C *enthält. (Da* $e \notin C'$ *haben wir eine gewisse Chance, daß* C' *ein Code ist, wer geht mit mir zum Wettbüro ?)*

Bemerkung 3.5 Eine Bemerkung zur Voraussetzung y unberandet und $y \in T(C^*)$: $y \notin T(C^*) \rightarrow \exists y'$, y' unberandet und $y' \notin T(C^*)$.

Beweis. Setze $V = X^* \setminus X^* y X^*$ und nun wissen wir anhand einer Übungsaufgabe (bereits gelöst?), daß Vy ein Präfix-Code ist. Annahme C' ist kein Code, d.h. $\exists w_i, w_i' \in C' w_1 \neq w_1'$ mit $w_1 \dots w_m = w_1' \dots w_n'$. Da C ein Code können nicht alle w_i, w_i' aus C sein, sei nun $w_k \notin C$ mit k minimal, d.h. $w_1, \dots, w_{k-1} \in Cy \sqsubset w_k$, damit folgt $a = w_1 \dots w_{k-1} y \sqsubset w_1' \dots w_n'$, $\rightarrow$ existiert ein l, l minimal mit $w_l' \in y(Uy)^* \rightarrow b = w_1' \dots w_{l-1}' y$ und $a \sqsubset b$, $a, b \in Vy$ ($y \notin T(C^*) \rightarrow C^* \subseteq V$), da Vy Präfix-Code folgt aber $a = b$ bzw. $k = l, w_i = w_i' \forall i < k$, da $w_1 \neq w_1'$ folgt $l = k = 0$, m.a.W. $w_1, w_1' \in y(Uy)^*$, d.h. $w_1 = yu_1 yu_2 \dots u_k y, w_2 = yu_1' yu_2' \dots u_k' y$ o.B.d.A. $k \leq l$. (Aus Bequemlichkeitsgründen sei k, l neu definiert.) Da $Uy \subseteq Vy$ ist somit Uy Präfix-Code und da $w_1 \sqsubseteq w_2$ folgt $u_i = u_i' i \leq k$ und aus $w_1 \neq w_1'$ folgt $k < l$. Damit folgt jedoch $u_{k+1}' y \dots u_l' y \sqsubseteq w_2 \dots w_m$. Sei nun wiederum i minimal gewählt bzgl der Eigenschaft $w_i \in y(Uy)^*$, beachte $i > 2$, da $y \not\sqsubseteq u_{k+1}' \rightarrow u_{k+1}' \sqsubset w_2 \dots w_{i-1} y$, dies ist jedoch gleichbedeutend mit $u_{k+1}' = w_2 \dots w_{i-1}$. Die letzte Behauptung widerspricht aber unserer Voraussetzung, die besagte nämlich $U \cap C^* = \emptyset$, d.h. die Annahme C' kein Code ist falsch. Es bleibt nun noch die Vollständigkeit von C' zu zeigen, d.h. $v \in X^* \rightarrow v \in T(C'^*)$. Jedes $V \in X^*$ hat eindeutige Faktorisierung $u_1 y u_2 \dots y u_n$ mit $u_i \in X^* \setminus X^* y X^* = C^* \cup U$, von den u_i seien $u_{i_1}, \dots, u_{i_k} \in C^*$ mit $i_j \leq i_k\, j \leq k$ die anderen seien aus U. Dann gilt $yvy = (yu_1 \dots yu_{i_1-1} y) u_{i_1} (yu_{i_1+1} y \dots yu_{i_2-1} y) u_{i_2} \dots u_{i_k} (yu_{i_k+1} y \dots u_n y) \rightarrow yvy \in y(Uy)^* \subseteq C'^*$. □

Satz 3.13 *Ist C ein maximaler Code, so ist C ein vollständiger Code.*

Beweis. Dieser Satz ist eine Folgerung des Satzes von Ehrenfeucht und Rozenberg, da falls C nicht vollständig, die Vervollständigung von C ein Code ist, der C enthält. □

Folgerung 3.14 *Ist $C \subseteq X^*$ ein Code, so gibt es einen vollständigen Code C' mit $C \subseteq C'$, ist C darüber hinaus reguläre Sprache, so gibt es auch einen regulären vollständigen Code, der C enthält. Zusammenfassung des Satzes von Ehrenfeucht und Rozenberg.*

Folgerung 3.15 *Ein Code ist vollständig $\iff$ der Code eine der beiden Eigenschaften erfüllt: der Code ist maximal oder dicht.*

Lemma 3.16 Es sei $L \subseteq X^*$ eine reguläre Sprache, dann sind die drei folgenden Bedingungen äquivalent

(1) L ist eine dichte Sprache

(2) es gibt ein Bernoulli - Maß $\mu : X^* \rightarrow [0,1]$ mit $\mu(L) = 1$

(3) für alle Bernoulli - Maße $\mu : X^* \rightarrow [0,1]$ ist $\mu(L) = 1$

Beweis. Da L reguläre Sprache gibt es ein $l \in \mathbb{N}$, so daß gilt $\forall v \in T(L) \exists u, w$ mit $|u|, |w| \leq l$ und $uvw \in L$, wähle, man kann ein l mit Hilfe eines DEA der die Sprache L akzeptiert angeben. Analog zu einem bereits in diesem Kapitel geführten Beweis zeigt man die folgendende Ungleichungskette : $\mu(T(L)) \leq \mu(L) \leq \frac{|X|}{\mu_{min}}^{2*l} * \mu(T(L))$ und damit folgen (3) aus (1), und trivialerweise folgt (2) aus (3). Es bleibt zu zeigen (1) folgt aus (2). Sei $\mu(L) = \infty \rightarrow \mu(T(L)) = \infty$, angenommen $T(L) \neq X^* \rightarrow \exists w \in X^*$ mit

$T(L) \subseteq M = X^* \setminus X^*wX^* \rightarrow \mu(T(L)) \leq \mu(M)$, aber es wurde bereits auf irgendeiner Seite des Manuskripts $\mu(M) < \infty$ gezeigt, Widerspruch. □

Folgerung 3.17 *Ein regulärer Code C ist mager, da für jedes Bernoulli - Maß $\mu : X^* \rightarrow [0,1] \mu(C) \leq 1$ nach dem Satz von MacMillan.*

Beispiel 3.2 Im folgenden wird gezeigt, daß aus der Vollständigkeit eines Codes nicht für jedes Bernoulli - Maß $\mu : X^* \rightarrow [0,1]$ $\mu(C) = 1$ folgen muß. Wir wissen bereits, daß ein solcher Code nicht regulär und dicht sein muß. Dafür ein erstes und einfaches Beispiel : Setze $X = \{a,b,c\} C = \cup_{n=0}^{\infty} a^n b X^n$, C ist Präfix-Code, und ebenso ist klar $T(C) = X^* = \cup_{n=0}^{\infty} X^n$. Definiert man nun μ durch $\mu(a) = \mu(b) = \mu(c) = \frac{1}{3}$, so folgt $\mu(C) = \frac{1}{2}$. Vorhang auf für eine ganze Reihe von Gegenbeispielen. Sei $l > 0, W, V \subseteq X^l, W \cup V = X^l, W \cap V = \emptyset$, definiere durch die folgende kontextfreie Grammatik

- S Startsymbol und $w \in W, v \in V$ Menge der zulässigen Endsymbole, $n \geq 1$ freier Parameter
- $S \rightarrow w\, w \in W$
- $S \rightarrow v S^n\, v \in V$

eine Sprache, die ein Präfix-Code ist. Es gilt offenbar $\mu(L(G)) = \mu(W) + \mu(V) * \mu(L(G))^n$, sowie $\mu(W) + \mu(V) = 1$. Zur Berechnung des Bernoulli-Maßes von $L(G)$ betrachte die folgende Darstellung von $L(G)$: $L(G) = \cup_{i=0}^{\infty} L_i\, L_0 = \emptyset, L_{i+1} = W \cup V L_i^n$. Offenbar gilt $L_i \subseteq L_j\, i \leq j$ und mit $\mu_i = \mu(L_i)$ folgt $\mu_{i+1} = \mu(W) + \mu(V) * \mu_i$. Aufgrund der Definition der μ_i gilt $\mu_i \leq \mu_j i \leq j$, d.h. die μ_i bilden eine positive monoton wachsende Folge reeller Zahlen, sei nun λ die kleinste nichtnegative Lösung der Gleichung $\lambda = \mu(W) + \mu(V) * \lambda^n$, dann folgt aus $\mu_0 = 0 \forall i\, \mu_i \leq \lambda$, (einfach aufschreiben), d.h. da $\lambda = 1$ Lösung der Gleichung, ist die Folge der μ_i beschränkt und somit konvergent, d.h. da der Grenzwert der Folge das Bernoulli-Maß von $L(G)$ war $\mu(L(G)) < \infty$. Betrachtung von $t = a + (1 - a) * t^n t \geq 0 \rightarrow t^n - t = a * (t^n - 1) \rightarrow \sum_{i=1}^{n-1} t^i = a * \sum_{i=0}^{n-1} t^i t < 1$, $t = 1$ uninteressant, da eine kleinere positive Lösung der Aufgabe gesucht ist. $\rightarrow (1 - a) * \sum_{i=1}^{n-1} t^i = a$ bzw. $\sum_{i=1}^{n-1} t^i = \frac{a}{1-a} \rightarrow t < 1 \iff \frac{a}{1-a} < n - 1 \iff a < \frac{n-1}{n}$, d.h. $\mu(L(G)) < 1 \iff \mu(W) < \frac{n-1}{n}$.

- Konstruktion eines Codes C mit einem Bernoulli - Maß μ_1 und $\mu_1(C) = 1$, sowie einem Bernoulli - Maß μ_2 und $\mu_2(C) \neq 1$. Definiere C wie oben bei setzen von $X = \{a,b\} W = \{a\}, V = \{b\}, n = 2$. Dann gilt $\mu_i(C) < 1 \iff \mu_i(a) < \frac{1}{2}$, setze $\mu_i(a)$ jeweils entsprechend.
- Konstruktion eines maximalen Codes C mit $\mu(C) < 1 \forall \mu$. Definiere C wie oben bei setzen von $X = \{a,b\} W = \{ab\}, V = \{aa, ba, bb\}, n = 2$. Es gilt $\forall \mu\, \mu(ab) = \mu(a) * (1 - \mu(a)) \leq \frac{1}{4} < \frac{1}{2} \rightarrow \mu(C) < 1$. Zu zeigen bleibt C ist maximal, dies sei dem geneigten Leser überlassen.

Lemma 3.18 Jeder (endliche) Präfix-Code ist in einem maximalen (endlichen) Präfix-Code enthalten. Letzterer maximaler Präfix-Code ist auch maximal als Code.

Beweis. Der Beweis dieses Lemmas ist eine Übungsaufgabe. □

Bemerkung 3.6 Im folgenden soll gezeigt werden, daß das von A. Markov 1967 angegebene Beispiel für einen endlichen Code, der in keinem endlichen maximalem Code enthalten ist, korrekt ist. Es sei vermerkt, das dieser A. Markov nichts mit den Markov-Ketten zu tun hat. Angenommen $C = \{b, ba, ba^5, a^2b\}$ sei in einem endlichen maximalen Code C' enthalten, dann ist C' ein vollständiger Code. Setze $l = max\{r|r = |w|w \in C'\} \rightarrow w = b^l a^{5l+4} b^l \in T(C')$, da C' Code ist das einzige Codewort der Form a^k gegeben durch a^5. (Angenommen es gibt ein weiteres Codewort dieser Form, z.b. a^r, betrachte Faktorisierung des Wortes a^{5r}, da C' Code folgt $r = 5$, im anderen Falle gäbe es zwei zulässige Zerlegungen und C' wäre kein Code.) Bei Betrachtung der Faktorisierung des Wortes w in C' folgt die Existenz zweier Codewörter w_1, w_2 in C', mit $w_1 = b^k a^m, w_2 = a^{m'} b^{k'}$, wobei die Relationen $k+n, k'+m' \leq l, n+n' \equiv 4 \mod 5$, d.h. $m = 5r+i, m' = 5r'+i', i+i' = 4, k, n' \neq 0$ gelten. Dies sieht man am besten anhand einer Grafik und anhand der Tatsache, daß bei einer Faktorisierung von w über C' l-mal das Codewort a^5 hintereinander auftreten muß, d.h. $b^k a^m (a^5)^l a^{m'} b^{k'} \in (C')^*$. Es folgt jedoch nun das C' kein Code sein kann.

- $(i, i') = (0, 4)$ $(ba)w_2 = (ba)(a^{5r'+4})(b) = (b)(a^5)^{r'+1}(b)^{k'}$
- $(i, i') = (1, 3)$ $w_1 = b^k a^{5r+1} = (b)^{k-1}(ba)(a^5)^r$
- $(i, i') = (2, 2)$ $w_2 = a^{5r+2} b^{k'} = (a^5)^r (a^2b)(b)^{r'-1}$
- $(i, i') = (3, 1)$ $w_1(a^2b) = (b^k a^{5r+3})(a^2b) = (b)^k (a^5)^{r+1}(b)$
- $(i, i') = (4, 0)$ $w_1(a^2b) = (b^k a^{5r+4})(a^2b) = (b)^{k-1}(ba)(a^5)^{r+1}(b)$

Bemerkung 3.7 Bis jetzt haben wir nur Klassen von Codes kennengelernt für die gilt: ein maximaler Code der Klasse ist bereits ein maximaler Code, z.B. gilt dies für die Klasse der Präfix-Codes, der Suffix-Codes, der regulären Codes und der mageren Codes. Es gibt jedoch auch Klassen von Codes für die dies nicht der Fall sein muß, z.b. die Klasse der sogenannten Bifixcodes, dies sind Codes, die sowohl Präfix- als auch Suffix-Codes sind. Als Aufgabe zum Einschlafen beweise man, daß der Code $C = \{a, ba^n b | n \in \mathbb{N}\}$ ein maximaler Bifixcode ist, der nicht maximal ist.

3.3 Decodierverzögerung

Definition 3.5 (Decodierverzögerung)

- Es sei $C \subseteq X^*$ ein Code und $w \in C$. Das Codewort w hat die Decodierverzögerung $m(w) \in \mathbb{N}$, falls aus $wv_1 \dots v_{m(w)} \sqsubseteq w'u$ mit $v_i, w' \in C$ und $u \in C^*$ stets $w = w'$ folgt.
- C hat endliche Decodierverzögerung, falls jedes $w \in C$ eine endliche Decodierverzögerung hat.
- C hat eine beschränkte Decodierverzögerung, falls es ein $(*)n \in \mathbb{N}$ gibt mit $m(w) \leq n \, \forall w \in C$ Falls n die kleinste natürliche Zahl ist, so spricht man von einem Code mit Decodierverzögerung n.

Bemerkung 3.8 Für endliche Codes fällt der Begriff beschränkte Decodierverzögerung und endliche Decodierverzögerung zusammen. In der Literatur variiert diese Definition etwas, manchmal wird der Wert m(w)+1 als Decodierverzögerung bezeichnet.

Beispiel 3.3 - ein Präfix-Code hat Decodierverzögerung 0

- $C = \{ba^n | n \in \mathbb{N}\}$ hat Decodierverzögerung 1
- $C = \{b(ba)^n | n \in \mathbb{N}\}$ hat Decodierverzögerung 2
- $C = \{ca, aba, c(aab)^n | n \in \mathbb{N}\}$ hat keine endliche Decodierverzögerung
- Der Code C dessen Codewörter induktiv durch $w_1 = b, w_i = w_i^i a^i b$ definiert sind, hat keine beschränkte Decodierverzögerung, da $m(w_i) \geq i - 1$, aber eine endliche Decodierverzögerung, da $m(w_i) \leq |w_{i+1}|$ ist.

Die jeweiligen Beweise für die Beispiele sind nicht kompliziert, jedoch schreibaufwendig.

Lemma 3.19 Wenn für jedes $w \in C$ jede mit w beginnende C-Kette eine Länge $\leq l_w$ hat, so hat C endliche Decodierverzögerung.

Beweis. Da die Existenz einer C-Kette $\{v_i\}_1^n \, v_i <_k v_{i+1} k = \{1, 2\}$ äquivalent zur Existenz von $w_k, u_k \in C$ mit $w_1 \dots w_{i-1} \sqsubseteq u_1 \dots u_j \sqsubseteq w_1 \dots w_i, w_1 \neq u_1, i + j = n$ ist, und jede C-Kette die mit w startet maximal die Länge l_w besitzt,d.h. $wu_2 \dots u_{l(w)-1} \subseteq w'$ folgt für die Decodierverzögerung von w im schlimmsten Falle $m(w) \leq l_w - 1$. □

Bemerkung 3.9 Die Frage nach der Gültigkeit der Umkehrung ist ein offenes Problem.

Lemma 3.20 Es sei $C \subseteq X^*$ ein Code mit Decodierverzögerung $m \in N$ und $y \notin T(C^*)$ ein unberandetes Wort, dann hat auch der Code $C \cup \{y\}$ die Decodierverzögerung m.

Beweis. Sei $wv_1 \dots v_m \sqsubseteq w'u$, zu zeigen ist $w = w'$.

- $w = y \rightarrow w' = y$
- $w, v_i \in C \rightarrow w' \in C'$, möglich ist $u \in C^x y(C \cup \{y\})^*$, dann ist $wv_1 \dots v_m \sqsubseteq w'u'y$ mit $u = u'yu''u' \in C^*$. Betrachten $wv_1 \dots v_m y$ und $w'u'y$ Element von C^*y, wissen $wv_1 \dots v_m y \subseteq w'u'y$, da C^*y Präfix-Code folgt Gleichheit und somit $w = w'$
- $y_j = yv_i \in C \, \forall i < j \rightarrow wv_1 \dots v_{j-1} y \sqsubseteq w'uu = u'yu''u' \in C^*$, analoges Schließen, wie im Fall vorher, liefert $w = w'$.

□

Folgerung 3.21 *Es sei C ein magerer Code mit beschränkter Decodierverzögerung, dann sind die folgenden beiden Bedingungen äquivalent:*

- *C ist maximaler Code*
- *C ist maximaler Code mit beschränkter Decodierverzögerung*

4 Kryptologie

Die Kryptologie, Teilgebiet der Codierungstheorie ist eines der klassischsten Gebiete der Mathematik, genauer gesagt der Zahlentheorie, sie beschäftigt sich mit dem Problem der Verschlüsselung einer Nachricht zwecks Geheimhaltung, bzw. mit dem Brechen der Verschlüsselung. Prinzipiell wird das folgende Schema betrachtet :
Sender $\rightarrow$ Angriff der Gegenpartei $\rightarrow$ Empfänger
Der Angriff kann passiver Art (Abhören) und aktiver Art (Nachrichten verfälschen) sein. Ziel der Kryptographie ist es nun einen Verschlüsselungsmechanismus $E : X^* \rightarrow Y^*(2^{Y^*})$ anzugeben, wobei E durchaus nichtdeterministisch sein kann, der das Klartextalphabet X in das Geheimalphabet Y überführt. Zu einem solchen E sollte es natürlich auch einen deterministischen Entschlüsselungsmechanismus $D : Y^* \rightarrow X^*$ geben mit

(1) $D(E(x)) = x \, \forall x \in X^*$

(2) $x \rightarrow E(x), y \rightarrow D(y)$ leicht berechenbar, falls E, D bekannt.

(3) $y \rightarrow D(y)$ (Idealfall) unmöglich falls D unbekannt, wobei die folgenden Fälle möglich sind

- Der Gegenpartei ist ein Geheimtext bekannt.
- Der Gegenpartei sind zwei Geheimtexte bekannt, wobei der gleiche Klartext mit dem gleichen Verfahren, aber mit unterschiedlichen Schlüsseln verschlüsselt wurde
- Ein Paar (Klartext, Geheimtext) der Gegenpartei bekannt ist.

Jeder der drei genannten Fälle ist in der Vergangenheit bereits aufgetreten.

4.1 Klassische Verfahren / Symmetrische Verfahren

4.1.1 Monoalphabetische Verfahren

Jeder Klartextbuchstabe wird einzeln verschlüsselt, d.h. durch einen Geheimtextbuchstaben ersetzt.

Caesar Codierung

Bei dieser Codierung handelt es sich um eine Monoalphabetische Substitution, d.h. gegeben ist $p : X \rightarrow X$, p ist eine Permutation und für $w \in X^* \, w = x_1 \dots x_r$, definiere $E_p(w) = p(x_1) \dots p(x_r)$. Die Caesar Codierung, angewendet von Caesar während seines gallischen Feldzuges, verwendete eine zyklische Permutation der Form : $a \rightarrow d, b \rightarrow e, c \rightarrow f, \dots x \rightarrow a, \dots$ Sei z.B. die zu codierende Nachricht *„Veni Vidi Vici“*, bei Anwendung des Verfahren geht sie über in *„yhqlylglylfl“*. Wie man sich leicht denken kann, ist ein solcher Code einfach zu knacken, z.B. Verwendung von sprachstatistischen Häufigkeitstabellen (ist das Auftreten des Buchstaben e am häufigsten, gefolgt von diversen anderen Vokalen). Eine genaue Beschreibung, wie ein solcher Code zu knacken ist, findet man z.B. in E.A. Poe's Geschichte *'Der Goldkäfer'*, oder in der Sherlock Holmes Geschichte *'Die tanzenden Männchen'*, beide Geschichten sind sehr zu empfehlen.

Skytale von Sparta

Bei diesem Verfahren handelt es sich um eine Permutation des Textes, Beispiele hierfür findet man in Rätselzeitschriften oft unter dem Stichwort Anagram. Zur Kodierung des

Textes schreibe man diesen Zeilenweise in ein Tableau mit n-Zeilen und m-Spalten. Der codierte Text ergibt sich, wenn der Inhalt des Tableaus spaltenweise aufgeschrieben wird. Sei z.B. $n = 3, m = 7$ und die zu codierende Nachricht *„Störe meine Kreise nicht“*, bei Anwendung des Verfahren geht sie über in *„sistneöenrkiercmeheit“*. Bei Anwendung des Verfahrens auf die codierte Nachricht und bei Verwendung eines Tableaus mit m-Zeilen und n-Spalten erhält man den Ausgangstext. Das originale Verfahren verwendete nicht Matrizen bzw. Tableaus, sondern einen Stock mit vorgegebener Länge und Dicke, die zu codierende Nachricht wurde im ersten Schritt auf einen Pergamentstreifen geschrieben, der Pergamentstreifen wurde um den Stock gewickelt, und der Text wurde nun reihenweise abgeschrieben. Ebenso wie eine Cäsar Codierung, ist auch dieses Verfahren relativ leicht zu knacken. z.B. Bestimmung der Wortlänge und Faktorisierung in Primfaktoren $\rightarrow$ Bestimmung aller möglichen Paare (n, m). Ebenso ist es auch möglich gewisse statistische Häufigkeiten auszunutzen.
Zur Erhöhung der Sicherheit können die folgenden Modifikationen verwendet werden :

- Kombination beider Verfahren
- Kleinschreibung, Weglassen der Interpunktion
- Rechtschreibfehler zur Störung der Sprachhäufigkeiten
- ...

Data Encryption Standart - DES
Ein wirkungsvolles Hilfsmittel zum Brechen der bisherigen monoalphabetischen Codes waren statische Aussagen über die Häufigkeit des Auftretens bestimmter Buchstaben. Um diese Aussagen wertlos zu machen, verwendet man bei DES eine künstliche Sprache.

- monoalphabetische Verschlüsselung über $X = \{0, 1\}^{64}$.
- der Schlüssel K_0 ist ein 64-Bit Wort (+8 Bit Parity Check), das Verfahren ist jedoch auch auf größere Schlüssel verallgemeinerbar.
- Ermittlung von $K_1, \ldots, K_{16}$ aus K_0 mit Hilfe einer nichtlinearen Vorschrift, die jedoch öffentlich zugänglich ist, siehe z.b. Salomaa 'Public Key Crypthographie' (Eine Bemerkung am Rande : Der Name Salomaa kommt aus dem finnischen und bedeutet 'friedliches Land', dies schreibe ich nur, um zu zeigen, daß man in dieser Vorlesung nicht nur etwas über Theoretische Informatik erfährt.)
- unterwerfe jedes x aus X einer Eingangspermutation P, und erhalte $P(x) = L_0R_0$, wobei die 32-Bit Wörter L_0, R_0 durch die ersten bzw. letzten 32-Bit des 64-Bit Wortes $P(x)$ gegeben sind.
- $L_i = R_{i-1}, R_i = L_{i-1} \oplus f(R_{i-1}, K_i) i = 1, \ldots 16$, mit wiederum öffentlich zugänglicher, aber natürlich nichtlinearer Funktion f, und $\oplus$ stehe für die XOR-Operation.
- $E(x) = P^{-1}(R_{16}L_{16})$ ist verschlüsselter Codebuchstabe.
- Entschlüsselung bei Kenntnis des Schlüssels K_0 erfolgt einfach Rückwärts : $R_{16}L_{16} = P(E(x))$, $R_{i-1} = L_i$, $L_{i-1} = R_i \oplus f(R_{i-1}, K_i)$, $x = P^{-1}(L_0R_0)$

Der Vorteil von DES besteht in der Schnelligkeit des Verschlüsselungs- bzw. Entschlüsselungvorgangs, und der mathematischen Eigenschaft, das kleine Änderungen des Ausgangstextes eine Abweichung von über 50% im Vergleich zum Ausgangstext hervorrufen. Der Nachteil von DES besteht in der Kürze der verwendeten Schlüssel, wenn man die heutige Rechentechnik in betracht zieht (Man wähle zufällig K_0, decodiere und begutachte das Ergebnis, bei Mißlingen neuer Versuch.), und in gewissen Ungereimtheiten bei Veröffentlichung des Verfahrens, d.h. man ist nicht 100% sicher ob es keine Falltür gibt, d.h. ob die Entwickler nicht einen Trick kennen, das Verfahren zu brechen.

4.1.2 Polyalphabetische Verfahren

Endliche Substitution
Sei $X = \{a_1, \ldots, a_n\}$ ein gegebenes Quellenalphabet und $P = \{p_1, \ldots, p_n\}$, die zugehörige Häufigkeitsverteilung, so definiere die Funktion $E : X \to 2^Y$ (2^Y steht für die Potenzmenge, d.h. die Menge aller Teilmengen von Y), die die folgenden Eigenschaften besitzen sollte :

(1) $E(a_i) \cap E(a_j) = \emptyset \; i \neq j$

(2) $\dfrac{|E(a_i)|}{|E(a_j)|} \sim \dfrac{p_i}{p_j}$

Die Verschlüsselung erfolgt nun dadurch, daß man einen Buchstaben $x \in X$ des Klartextes durch ein zufällig gewähltes y aus $E(x)$ ersetzt. Die Eigenschaft (1) garantiert die eindeutige Decodierung, die Eigenschaft (2) bewirkt eine annähernde Gleichverteilung der im codierten Text auftretenden Buchstaben. Um einen solcherart konstruierten Code zu brechen, untersucht man die Auftrittshäufigkeit von Buchstabenpaaren, -tripeln usw., die man dann wiederum sprachstatisch auszuwerten versucht.

Vigenere Codierung
Die Vigenere Codierung wurde 1584 von dem französischen Diplomaten Vigenere entwickelt und galt bis in die Mitte des letzten Jahrhunderts als sicherste aller bekannten Codierungen, sie wurde um 186? von einem preußischen Offizier namens Kasiski gebrochen, der amerikanische Mathematiker Friedman veröffentlichte 1921 eine verbesserte Methode diese Codierung erfolgreich zu brechen.
Beschreibung des Codierungsvorganges :

- Wahl des Schlüsselwortes, z.B. VIGENERE
- Schreibe das Schlüsselwort am laufenden Band unter den zu verschlüsselnden Text, z.B. Le roi est mort, vive le roi ! (Hoffentlich spricht hier niemand Französisch, er könnte an Herzschlag sterben, aufgrund maßloser Aufregung darüber, wie an dieser Stelle mit der göttlichen Sprache der Franzosen umgegangen wird.)
- LE ROI EST MORT VIVE LE ROI
- VI GEN ERE VIGE NERE VI GEN
- Addition der Buchstaben modulo 26 $\rightarrow$
- DN YTW JKY LXYY JNDJ HN YTW

- Decodierung erfolgt durch Aufschreiben des Geheimtextes, darunter Schreiben des Schlüsselwortes und Subtraktion der Buchstaben modulo 26

Zur Kryptanalyse : Beim Versuch diese Codierung zu brechen, fällt auf, daß bei Kenntnis der Schlüsselwortlänge n, die Vigenere Codierung in n Cäsar Codierungen zerfällt, mit dem Schlüsselwort VIGENERE wird jeder $(8k+1)$te Buchstabe einer Cäsar Codierung mit Parameter 22 unterzogen, da V der 22. Buchstabe des Alphabets ist, m.a.W. kennt man n ist die Codierung, so gut wie gebrochen. Kasiski bestimmte n mittels folgender Methode : Er suchte nach langen Teilworten im Geheimtext, die sich oft wiederholten. (die Länge der Teilworte ≥ 3), faktorisierte diese Längen und bestimmte den größten gemeinsamen Teiler der Längen, der sich wiederholenden Teilworte, dieser ggt war seine Schätzung für n, brach die zugehörigen Cäsar Codierungen und hoffte einen vernünftigen Klartext zu erhalten. Diese Methode ist relativ erfolgreich, aber besonders mühsam ohne die Hilfe eines Computers. Friedman's Verbesserung bestand in der Angabe einer Methode die zur Abschätzung für die Größenordnung von n genutzt werden konnte. Hierfür führte er den Begriff des Koinzidenzindex ein.

Definition 4.1 (Koinzidenzindex) Sei $Q = (X, P)$ eine Quelle mit dem Quellenalphabet $X = \{a_1, \dots\}$, wobei p_i, die dem Buchstaben a_i, zugeordnete relative Häufigkeit ist. Für $w \in X^*$ setze

$$K(w) = \sum_{i=1}^{|X|} \frac{n_i(n_i - 1)}{n(n-1)}$$

wobei $n = |w|$ und n_i die Anzahl des Buchstaben a_i im Wort w. Der Koinzidenzindex einer Quelle Q ergibt sich zu

$$K_Q = \sum_{i=1}^{|X|} p_i^2$$

Bemerkung 4.1 Der Wert $K(w)$ gibt an, wie groß die Wahrscheinlichkeit ist, daß zwei unterschiedlich gewählte Buchstaben aus w identisch sind. Ist das Wort w von der Quelle Q erzeugt und $|w| \gg 1$, so gilt $K(w) \simeq K_Q$. Nimmt man als Quelle die deutsche Sprache, so beträgt $K_{deutsch} = 0.076$, im Falle der englischen Sprache ist $K_{englisch} = 0.066$. K_Q ist minimal, wenn die Quelle gleichverteilt ist, und es gilt $K_{gleich} = 0.0385$

Folgerung 4.1 *Der Koinzidenzindex ist invariant gegenüber Monoalphabetischen Codierungsverfahren.*

Diese Eigenschaft läßt sich nun im Fall der Vigenere Codierung wie folgt ausnutzen : Angenommen die Schlüsselwortlänge sei l, schreibe Kryptotext Y mit $|y| = n$ in ein Tableau mit n Spalten zeilenweise auf. Die einzelnen Spalten sind nun Monoalphabetisch codiert, d.h. der Koinzidenzindex eines Spaltenwortes ist in etwa identisch mit dem Koinzidenzindex K_Q der jeweiligen Originalsprache. Die Wahrscheinlichkeit, daß zwei Buchstaben aus **verschiedenen** Spalten gleich sind, entspricht in etwa K_{gleich}, der Idealfall wäre ein Zufallswort als Schlüsselwort. Der Koinzidenzindex des Schlüsselwortlänge läßt sich nun in etwa durch

$$K(Y) \simeq \frac{K_Q * A_{gleich} + K_{gleich} * A_{verschieden}}{n(n-1)/2}$$

berechnen, wobei $n*(n-1)/2$ die Anzahl der möglichen verschiedenen Paare aus Y angibt. $A_{gleich} = (n/l*(n/l-1))/2$ ist die Anzahl zwei verschiedene Buchstaben in der gleichen Spalte zu wählen, und $A_{verschieden} = n/2*(n-n/l)$ ist Anzahl zwei Buchstaben in zwei verschiedenen Spalten zu wählen. Damit ergibt sich als Schätzwert für die Schlüsselwortlänge

$$l \simeq \frac{(K_Q - K_{gleich})*n}{n*(K(Y)-K_{gleich}} + (K_Q - K(Y))$$

Diese Schätzung und die Methode von Kasiski liefert Kandidaten für die Schlüsselwortlänge.

ONE TIME PAD

Ausgehend von der Vigenere Codierung sind weitere Verfahren entwickelt worden. Die Sicherheit des Verfahrens wird erhöht bei Verwendung von langen mehr oder weniger gleichverteilten Schlüsselwörtern. Eine Methode besteht in der Verwendung von Buchtexten, d.h. alles was ab einer gewissen Stelle in einem Buch auftaucht wird als Schlüsselwort verwendet, d.h. Schlüsselwortlänge ist gleich Klartextlänge. Eine andere Methode wurde 1917 von Vernam entwickelt. $X = \{0,1\}$ Als Schlüssel wird eine gleichverteilte und unabhängig Zufallsfolge verwendet. Sei $K = k_1k_2\ldots k_n, W = x_1x_2\ldots x_n \rightarrow E(W) = K \oplus W$, wobei $\oplus$ die XOR-Operation ist. Es folgt sofort, das $E(W)$ eine unabhängige und gleichverteilte Zufallsfolge ist, und damit ist die Codierung im kryptologischen Sinne sicher. Die Decodierung erfolgt durch $D(E(W)) = E(W) \oplus K$. Den Namen erhielt das Verfahren durch das Wegwerfen des Schlüssel nach Benutzung. Es ergibt sich nun das Problem der Schlüsselübergabe :

- unproblematisch falls nur in Ausnahmefällen gesendet werden soll und Schlüsselübergabe lange vor dem Senden möglich
- Verwendung von Pseudo-Primzahlen, d.h. nur die Anfangsfolge ist nötig zu übermitteln, weitere Glieder der Folge werden auf deterministische Weise erzeugt. Die Erzeugung der Pseudo-Primzahlen erfolgte meist mittels eines Apparates, ein Beispiel ist die ENIGMA, die von den Deutschen im 2.Weltkrieg verwendet und von britischen Mathematikern (z.B. Turing) geknackt wurde.

Der Vorteil von symmetrischen Verfahren ist ihre Schnelligkeit.

4.2 Asymmetrische Verfahren / Public Key Kryptographie

4.2.1 Darstellung des Grundgedanken asymmetrischer Verfahren

4.2.2 Zahlentheoretische Grundlagen

- Es gilt der Fundamentalsatz der Arithmetik über die Eindeutigkeit der Primfaktorisierung einer natürlichen Zahl.
- $r = ggt(a,b) \iff r|a, r|b$, und, falls $s|a, s|b \rightarrow s \leq r$ Der $ggt(a,b)$ kann durch den Euklidischen Algorithmus bestimmt werden in $O(\log^3(a))$ falls $a \geq b$. Der Euklidische Algorithmus liefert weiterhin eine Darstellung des $ggt(a,b)$ als Linearkombination von a und b, die Werte $z_1, z_2 \in \mathbb{Z}$ mit $ggt(a,b) = z_1*a + z_2*b$.
- $a \equiv b \mod m \iff \exists k \in \mathbb{Z} a = b + k*m$, die folgende abkürzende Schreibweise wird vereinbart : $b = (a \mod m) \iff a \equiv b \mod m$ und $0 \leq b < m$. Es sei

hinzugefügt, daß es eine Unzahl von Schreibweisen gibt, die alle irgendwann und irgendwo in der Literatur auftauchen.

Definition 4.2 (Eulersche φ-Funktion) Setze $M(n) = \{a | ggt(a,n) = 1, 0 < a < n\}$ $n \in \mathbb{N}$, so heißt die durch $\varphi : \mathbb{N} \to \mathbb{N}, \varphi(n) = |M(n)|$ definierte Funktion Eulersche φ-Funktion, m.a.W. $\varphi(n)$ ist gleich der Anzahl der zu n teilerfremden Zahlen.

Satz 4.2 (Kleiner Satz von Fermat) *Es sei p eine Primzahl, dann gilt für alle $a \in \mathbb{N} a^p \equiv a \mod p$, gilt außerdem $p \nmid a \to a^{p-1} \equiv 1 \mod p$.*

Beweis. Sei $ggt(p,a) = 1$, d.h $p \nmid a$, man betrachte die Menge $\{0a, 1a, \ldots, (p-1)a\}$. Angenommen $ia \equiv ja \mod p \to (i-j)a \equiv 0 \mod p \to p|(i-j)a$, da $p \nmid a$ folgt $p|(i-j)$, und aus $-p < i-j < p$ folgt $i = j$, d.h $\forall i \in \{0,1,\ldots,(p-1)\} \exists j \in \{0,1,\ldots,(p-1)\}$ mit $i \equiv ja \mod p \to \prod(ia) = a^{p-1}(p-1)! \equiv (p-1)! \mod p$, d.h. $(a^{p-1}-1)((p-1)!) \equiv 0 \mod p$, da $p \nmid (p-1)! \to p|(a^{p-1}-1)$, d.h $a^{p-1} \equiv 1 \mod p$, insbesondere folgt somit auch $a^p \equiv a \mod p$. Im Falle $ggt(p,a) = p$ gilt trivialerweise $a^p \equiv 0 \equiv a \mod p$. □

Satz 4.3 (Chinesischer Restsatz) *Es seien $m_1, \ldots, m_k$ paarweise verschiedene und teilerfremde natürliche Zahlen, dann hat das System $a \equiv a_i \mod m_i i = 1, \ldots k$ für beliebige a_i eine bis auf Kongruenz modulo $M = \prod m_i$ eindeutig bestimmte Lösung.*

Beweis.

- Existenz einer Lösung
  Setze $M_i = M/m_i \to ggt(M_i, m_i) = 1 \to \exists N_i N_i M_i \equiv 1 \mod m_i$. Zur Bestimmung des N_i benutze Euklidischen Algorithmus, dieser liefert $r, s \in Z$ mit $rM_i + sm_i = 1$, setze $N_i = r$. Setze $x_i = a_i N_i M_i$ bzw. $x = \sum x_i \to x \equiv x_i \equiv a_i \mod m_i$, d.h x ist Lösung unseres System von Kongruenzen.

- Eindeutigkeit der Lösung
  Seien x_1, x_2 Lösungen des System, dann gilt für $y = x_1 - x_2 y \equiv 0 \mod m_i \forall m_i \to m_i | y$, da $ggt(m_i, m_j) = 1$ für $i \neq j$ folgt $M \mid y$, bzw. $y \equiv 0 \mod M$, bzw $x_1 \equiv x_2 \mod M$.

□

Satz 4.4 (Eigenschaften und Darstellung der φ-Funktion)

- *p Primzahl* $\to \varphi(p^\alpha) = p^\alpha - p^{\alpha-1}\alpha \geq 1$
- $ggt(m,n) = 1 \to \varphi(mn) = \varphi(m) * \varphi(n)$

Folgerung 4.5 *Sei mit $n = \prod p_i^{\alpha_i}$ die Primfaktorisierung gegeben, $p_i \neq p_j$ $i \neq j$, $\alpha_i \geq 1$, so gilt $\varphi(n) = \prod(p_i^{\alpha_i} - p_i^{\alpha_i - 1}) = n * \prod(1 - 1/p_i)$*

Beweis.

- Ist $0 \leq a \leq p^\alpha$, so gilt $ggt(a, p^\alpha) = 1 \iff p \nmid a$, in der Menge $\{0, \ldots p^\alpha - 1\}$ gibt es genau $p^\alpha/p = p^{\alpha-1}$ Zahlen, die durch p teilbar sind, d.h. $p^\alpha - p^{\alpha-1}$ sind es nicht, d.h. $\varphi(p^\alpha = p^\alpha - p^{\alpha-1}$ per Definition der φ-Funktion.

- Für $j \in \{0, \ldots, mn-1\}$ betrachte System $j_1 \equiv j \mod m, j_2 \equiv j \mod n$, da m, n teilerfremd liefert der Chinesische Restsatz für jedes Paar $(j_1, j_2) 0 \leq j_1 \leq m-1, 0 \leq j_2 \leq n-1$ genau eine Lösung j des Systems mit $0 \leq j \leq mn-1$, d.h. zu jedem Paar mit $ggt(j_1, m) = 1, ggt(j_2, n) = 1$ existiert genau ein j mit $ggt(j, mn) = 1$ und umgekehrt, d.h. $\varphi(mn) = \varphi(m) * \varphi(n)$

□

Satz 4.6 (Satz von Fermat und Euler) *Sei $a \in N, ggt(a, m) = 1$, dann gilt $a^{\varphi(m)} \equiv 1 \mod m$.*

Bemerkung 4.2 Dies ist eine Verallgemeinerung des kleinen Satzes von Fermat, der nur eine Aussage macht für $m = p$ Primzahl.

Beweis. Im ersten Schritt wird die Aussage für $m = p^\alpha$, p Primzahl und $\alpha \geq 1$ bewiesen mittels Induktion.

(IA) $\alpha = 1$, Aussage ist wahr, da $\varphi(p) = p - 1$ und der kleine Satz von Fermat gilt.

(IV) Satz gelte nun für $\alpha \geq 1$.

(IB) Aus der Gültigkeit des Satzes für α folgt Gültigkeit des Satzes für $\alpha + 1$.

(IS) Da $\varphi(p^\alpha) = p^\alpha - p^{\alpha-1}$, gilt $\varphi(p^{\alpha+1}) = p * \varphi(p^\alpha)$, mithin $a^{\varphi(p^{\alpha+1})} = (a^{\varphi(p^\alpha)})^p = (1 + r * p^\alpha)$ nach Induktionsvoraussetzung, d.h. $a^{\varphi(p^{\alpha+1})} = 1 + \sum_{v=1}^{p} \binom{p}{v} * (rp^\alpha)^v$ (Binominal Lehrsatz). Da $p | \binom{p}{v}$ mit $1 \geq v \geq p-1$ folgt die Behauptung.

Seien nun $n_1, n_2 \in \mathbb{N}$ mit $ggt(n_1, n_2) = 1, a^{\varphi(n_i)} \equiv 1 \mod n_i$, dann gilt $a^{\varphi(n_1 n_2)} = a^{\varphi(n_1)\varphi(n_2)}$, d.h. nach Anwendung bekannter Rechenregeln

$$a^{\varphi(n_1 n_2)} \equiv 1 \mod n_i \rightarrow a^{\varphi(n_1 n_2)} \equiv 1 \mod n_1 n_2$$

Da jede natürliche Zahl ein Produkt von zueinander teilerfremden Primzahlpotenzen ist, folgt die Behauptung. □

Bemerkung 4.3 Falls p Primzahl, so gilt für alle Zahlen a $a^{\varphi(p)+1} \equiv a \mod p$, dies ist falsch für ein beliebiges Paar (a, n), z.B. $a = 2, m = 12 \rightarrow 2^{\varphi(12)+1} = 2^{4+1} \equiv 8 \mod 12$. Es gilt jedoch der folgende Satz :

Satz 4.7 *Ist $n = \prod p_i$ mit p_i Primzahl und $p_i \neq p_j$ $i \neq j$, dann gilt für jede natürliche Zahl, aus $m \equiv 1 \mod \varphi(n)$, d.h. insbesondere für $m = \varphi(n) + 1$, folgt $a^m \equiv 1 \mod n$.*

Beweis. Es ist ausreichend aufgrund der einfachen Struktur von n, wenn $a^m \equiv 1 \mod p_i$ gilt. Aufgrund der Voraussetzungen kann m geschrieben werden als $m = r_1 * \varphi(n) + 1$. Sei nun p_i beliebig aber fest gewählt, damit $\varphi(n) = r_2 * (p_i - 1)$.

- $p_i \nmid a$
  Setze $m = r_1 * \varphi(n) + 1$ und $\varphi(n) = r_2 * (p_i - 1)$, mit $r_1, r_2 \in \mathbb{N}$ dann folgt $a^m = a^{r_1 * \varphi(n)+1} = a * (a^{p_i - 1})^{r_1 * r_2} \equiv a \mod p_i$ nach dem kleinen Satz von Fermat.

- $p_i | a$
  Es gilt nun trivialerweise $a^m \equiv 0 \equiv a \mod p_i$

□

Modulare Potenzierung

Zur effizienten Berechnung von $a \equiv b^n \mod m$ für m, n relativ groß, verwendet man i.A. die modulare Potenzierung. Sei $n = \sum_{k=0}^{r} n_k * 2^k a_k \in \{0, 1\}$. Setze $a_0 = 1$ falls $n_0 = 0$, bzw. $a_0 = b$ falls $n_0 = 1$, weiter setze $b_0 = (b \mod m)$. Dann berechne b_i durch $b_i = (b_{i-1}^2 \mod m)$. Setze a_i gleich a_{i-1} falls $n_i = 0$ bzw. gleich $(a_{i-1} * b_i \mod m)$ falls $n_i = 1$. Es gilt nun $a_r \equiv b^n \mod m$. Diese kann mittels vollständiger Induktion bewiesen werden. Der Aufwand berechnet sich aus $r = \log(n)$ Multiplikationen von Zahlen der Länge $\log(m)$, d.h. bei Verwendung des einfachsten Algorithmus zur Berechnung eines Produktes zweier natürlicher Zahlen gilt, daß der Aufwand von der Größenordnung $O(\log(n) * \log^2(m))$ ist.

4.3 Primzahltests

Im folgenden Abschnitt soll es um die Frage gehen, wie kann man effizient eine vorgegebene natürliche Zahl p auf Primalität testen. Eine Möglichkeit dies zu bewerkstelligen, ist das Sieb des Eratosthenes, d.h teste alle Primzahlen $q \leq \sqrt(p)$ ob $q \nmid p$. Ein solches Vorgehen ist natürlich recht aufwendig. Da es jedoch keine besseren Algorithmen gibt, die die Primaltität von p mit Sicherheit feststellen, verwendet man i.A. probabilistische Tests, d.h. nach Ablauf des Tests ist p höchstwahrscheinlich eine Primzahl.

Satz 4.8 *Für $n \in \mathbb{N}$ sei $A(n) = \{a | 1 \leq a \leq n-1, ggt(a, n) = 1\}$, dann ist $a^{n-1} \equiv 1 \mod n$ entweder für alle $a \in A(n)$ erfüllt oder höchstens für die Hälfte der Elemente aus $A(n)$.*

Beweis. Setze $Z(n) = \{z | z \in A(n), z^{n-1} \equiv 1 \mod n\}$. Sei $a \in A(n) \setminus Z(n)$, setze $a_i = (a * z_i \mod n) \forall z_i \in Z(n)$.

(1) $a_i \in A(n)$

(2) $a_i \notin Z(n)$, da $a_i^{n-1} \equiv a^{n-1} z_i^{n-1} \equiv a^{n-1} \not\equiv 1 \mod n$,

(3) $a_i = a_j \rightarrow a * (z_i - z_j) \equiv 0 \mod n \rightarrow Z_i = z_j$

Falls es ein $a \in A(n) \setminus Z(n)$ gibt, so folgt aus (2) und (3) sofort $|A(n) \setminus Z(n)| \geq |Z(n)|$, d.h. die Behauptung des Satzes. □

Auf der Grundlage dieses letzten Satzes und des kleines Satzes von Fermat arbeitet der Fermat - Primzahl Test.

Fermat-Test

Primzahltest von n bzgl. a

- Bestimmung von $ggt(a, n)$, falls $ggt(a, n) \neq 1$, so folgt n ist keine Primzahl
- Bestimmung von $a^{n-1} \mod n$, falls $a^{n-1} \not\equiv 1 \mod n$, so folgt n ist keine Primzahl, andernfalls hat n den Test bzgl. a bestanden, d.h. n ist Pseudoprimzahl bzgl. a.

Besteht eine Zahl n den Test k-mal bzgl. zufällig gezogenen a_i, so gilt mit obigem Satz $A(n) = Z(n)$ mit Wahrscheinlichkeit von mindestens $1 - 1/2^k$, d.h.

(a) n ist eine Primzahl

(b) n ist eine Carmichael - Zahl, d.h. eine zusammengesetzte Zahl mit $A(n) = Z(n)$, ein Beispiel für eine solche Zahl ist $543 = 3 * 7 * 23$, sie ist die kleinste natürliche Carmichael - Zahl.

Der Fall (b) ist relativ selten und sollte jemand eine neue Carmichael - Zahl finden, so ist ihm die Aufnahme in den mathematischen Olymp sicher. Der Beweis der Existenz unendlich vieler Carmichael - Zahlen wurde erst vor kurzem erbracht. Da Fall (b) so selten ist, ist der Fermat Test in der Praxis verwendbar. Es gibt andere Verfahren ohne Ausnahme (b) (Solovay - Strassen - Test, Miller - Rabin - Test), letzterer Test besitzt sogar eine höhere Konvergenzrate gegen die Wahrscheinlichkeit 1, und könnte zu einem deterministischen Test auf Primalität modifiziert werden, mit einer polynomialer Laufzeit bzgl. der Länge der Zahl n, falls die noch unbewiesene Riemannsche Vermutung wahr ist.

Satz 4.9 (Satz von C.F.Gauß) *Sei $\pi(n)$ die Anzahl der Primzahlen kleiner oder gleich n, dann gilt $\lim_{n\to\infty} \frac{\pi(n) * \ln(n)}{n} = 1$, m.a.W. $\pi(n) \simeq \frac{n}{\ln(n)}$*

Bemerkung 4.4 Mit Hilfe dieses Satzes läßt sich z.B. die Anzahl der 150-ziffrigen Primzahlen zu $1.4 * 10^{148}$ abschätzen, d.h. ca. jede 50. ist eine Primzahl, da an Primzahlen, die beim RSA-Verfahren verwendet werden noch weitere Forderungen gestellt sind, sind ca. 1000 Primzahl-Tests nötig zum Auffinden einer geeigneten Primzahl.

4.4 RSA - Verfahren

Das RSA - Verfahren wurde entwickelt von **R**ivest, **S**hamir und **A**delmann und veröffentlicht im Jahre des Herrn 1978. Nutzer A wähle zwei Primzahlen p_A, q_A, berechnet $n_A = p_A * q_A$, $\varphi(n_A) = (p_A - 1) * (q_A - 1)$., wählt einen Verschlüsselungsexponenten e_A, wobei e_A eine relative kleine Primzahl mit $ggt(\varphi(n_A), e_A) = 1$ gelten muß, beim Testen dieser Eigenschaft mittels des Euklidischen Algorithmus berechnet er d_A mit $e_A * d_A \equiv 1 \mod \varphi(n_A)$. Nun veröffentlicht er das Paar (n_A, e_A), während er $p_A, q_A, \varphi(n_A), d_A$ geheim hält, in der Praxis kann man sogar jede Aufzeichnung über p_A, q_A vernichten, die im weiteren nicht mehr benötigt werden. Die Verschlüsselung einer Nachricht m erfolgt nun durch Codierung von m als Folge von Blöcken, wobei jeder Block einer Zahl m_i entspricht, mit $m_i < n_A$. Die Verschlüsselungsvorschrift lautet nun $E_A(m_i) = m_i^{e_A} \mod n_A$ und die Entschlüsselungsvorschrift $D_A(y_j) = y^{d_A} \mod n_A$. Es gilt $D_A(E_A(m_i)) = m_i^{e_A * d_A} \mod n_A = m_i$, da $m_i < n_A$ und $e_A * d_A \equiv 1 \mod \varphi(n_A)$ (Satz von Fermat und Euler). Der einzige bis heute bekannte Weg zum Brechen der Kryptanalyse besteht in der Faktorisierung von n_A, es ist unbekannt ob die Komplexität dieses Problems polynomial oder exponential ist. Die besten Laufzeiten im Jahre 1994 waren $\exp(O((\log(n) * \log(\log(n)))^{\frac{1}{2}}))$, bzw. $\exp(O(\log(n)^{\frac{1}{3}} * (\log(\log(n)))^{\frac{2}{3}}))$, m.a.W. eine Faktorisierung von Zahlen mit 150 Stellen ist im Bereich des möglichen. Zur Erhöhung der Sicherheit stellt man die folgenden Forderungen an die frei wählbaren Primzahlen p_A, q_A, e_A :

(1) Die Primzahlen p_A, q_A seien Zahlen in der Größenordnung von 10^{150}

(2) $|p_A - q_A| \gg 1$, denn aus $n = p_A * q_A \to n = \frac{p_A+q_A}{2}^2 - \frac{p_A-q_A}{2}^2$, d.h. falls $|p_A - q_A| \simeq 1$, so versucht man n als Differenz von Quadraten darzustellen.

(3) $ggt(p_A, q_A - 1)$ sollte klein sein, sonst existiert kleiner Entschlüsselungsexponent e_A

(4) $\varphi(n_A)$ sollte große Primfaktoren enthalten

Bemerkung 4.5 e_A sollte klein, aber nicht zu klein gewählt werden. z.B. $e_A = e_B = e_C = 3$, sei m eine Nachricht die an A, B, C zugleich gesendet wird, man empfängt $m^3 \equiv n_i i = A, B, C$, Nach Chinesischem Restklassensatz gibt es m_0 mit $m_0 \equiv m^3 \mod m_i, m_0 \leq n_A n_B n_C \rightarrow m_0 \equiv m^3 \mod n_A n_B n_C$, da $m < n_i$ folgt $m = n^3$, d.h. $n = \sqrt[3]{m}$.

4.5 Elektronische Unterschriften

5 Kolmogorov Komplexität

Die Anfänge der Theorie der Kolmogorov Komplexität liegen im Jahre 1919 als **von Mieses** versuchte zu definieren was eine zufällige Folge ist. Er versuchte dies über die „Beschreibungskomplexität“ der Folge und sagte sie wäre zufällig, falls es keine Muster bzw. keine Gesetzmäßigkeiten gäbe. **Kolmogorov** beschritt 1935 einen anderen Weg die Wahrscheinlichkeitstheorie zu begründen und gab eine mengentheoretische Begründung der Wahrscheinlichkeitsrechnung. Dieser Weg läßt jedoch die Frage: Was ist ‘zufällig‘ ? unbeantwortet. Beispiele für zufällige $0-1-$Folgen :

(a) $00000\ldots0$ - n-mal Ziffer 0

(b) $0101101001011010\ldots$ - n-mal 01011010

(c) 00101001100110001110101O - für mich eine Folge ohne Gesetzmäßigkeit

Die Theorie der Kolmogorov Komplexität versucht die Fälle (a),(b) vom Fall (c) zu trennen und nur die Folge (c) als zufällige Folge zu bezeichnen. Die Väter der Kolmogoroff Komplexität sind Solomonoff (1964), Kolmogoroff (1965), Chaitin (1966).

Definition 5.1 Sei $U : X^* \times X^* \rightarrow X^*$ eine partiell - rekursive Fkt, d.h. $U(w, v)$ nicht für alle Paare (w, v) definiert.

$$K_U(w|v) = \inf\{|\pi||U(\pi, v) = w\}$$

Bemerkung 5.1 $K_U(w|v)$ beschreibt die Mindestgröße der Eingabe zur Erzeugung eines Wortes w bei Verwendung des Algorithmuses U, falls die Zusatzinformation v dem Algorithmus U bekannt ist. Man kann U auch als Interpreter des Programmes π auffassen, d.h. $K_U(w|v)$ wäre dann die Mindestlänge eines Programmes, zur Erzeugung des Wortes w bei Verwendung des Interpreters U.

Beispiel 5.1 (1) $U_1(\pi, v) = w \rightarrow K_{U_1}(w|v) = 0, K_{U_1}(w|v) = \infty$ falls $w' \neq w$

(2) $U_2(\pi, v) = \pi \rightarrow K_{U_2}(w|v) = |w|$

(3) $U_3(\pi, v) = v \rightarrow K_{U_3}(w|v) = 0\ (w = v), = \infty\ (w \neq v)$

(4) Wörter mit wenig Einsen, bezeichne $|w|_1$ die Anzahl der Einsen im Wort w, so setze

$$L_\lambda = \{w|w \in \{0, 1\}^*, |w|_1 \leq \lambda * |w|\}$$

Es gilt

$$|L_\lambda \cap \{0,1\}^n| = \sum_{i=0}^{\lfloor \lambda * n \rfloor} \binom{n}{i} \simeq 2^{n * h_2(\lambda)} \geq 0$$

mit $h_2(\lambda) = -\lambda * \log_2(\lambda) - (1-\lambda) * \log_2(1-\lambda)$. $U_\lambda(\pi, v) = \pi$-tes Wort in $L_\lambda \cap \{0,1\}^{|r|}$ in der quasilexikographischen. Ordnungsrelation über X^*. $K_{U_\lambda}(w|v) \leq h_2(\lambda) * |w|$ $(|w| = |v|), = \infty$ $(|w| \neq |v|)$.

Bemerkung 5.2 Die Funktion $h_2(\lambda)$ in Beispiel (4) ist die Entropie - Funktion, die bereits bei der Shannon - bzw. Huffman Codierung eine Rolle gespielt hat.

Satz 5.1 *Es gibt eine partiell - rekursive Fkt. $U : X^* \times X^* \to X^*$ für welche gilt : Für jede partiell - rekursive Fkt. $\varphi : X^* \times X^* \to X^*$ existiert eine Konstante c_φ mit $K_U(w,v) \leq K_\varphi(w,v) + c_\varphi \quad \forall (w,v)$.*

Beweis. Es gibt eine universelle Turingmaschine, die die Fkt. $U(Code(\varphi)\pi, v) = \varphi(\pi, v)$ berechnet. Wenn $\varphi(\pi, v) = w \to U(Code(\varphi)\pi, v) = w \to K_U(w|v) \leq K_\varphi(w|v) + |Code(\varphi)|$ $\forall (w,v)$. □

Folgerung 5.2 *(1) Es gilt $K_U(w|v) < \infty$, verwende für φ im obigen Satz die Fkt. U_1.*

(2) Es gilt $K_U(w|v) \leq |w| + c$, verwende für φ im obigen Satz die Fkt U_2.

(3) Es seien U, U' universelle partiell rekursive. Fkt., dann existiert eine Konstante c, so daß $|K_U(w|v) - K_{U'}(w|v)| \leq c$ $\forall (w,v)$.

Folgerung 5.3 *Für jedes $n \in \mathbb{N}$ gibt es ein $w \in X^n$ mit $K_U(w|v) \geq n$.*

Beweis. Die Anzahl der Programme π, $|\pi| < n$ ist kleiner als die Anzahl der Wörter mit Länge $|\pi|$ (geometrische Reihe), d.h. nicht alle Wörter mit Länge n können von Programmen erzeugt werden mit Länge echt kleiner n. □

Folgerung 5.4 *Es sei $f : X^* \to X^*$ eine rekursive Fkt. Dann gibt es eine Konstante c_f, so daß $K_U(f(w)|v) \leq K_U(w,v) + c_f$ für alle Paare (w,v).*

Beweis. $f : X^* \to X^*, U : X^* \times X^* \to X^* \to f \circ U :\to X^* \times X^* \to X^*$ ist partiell rekursive Fkt. $f(w) = f(U(u', v))$ mit u' gegeben durch $U(u', v) = w \to K_{f \circ U}(f(w)|v) \leq K_U(w|v)$, und mit obigem Satz $K_U(f(w)|v) \leq K_{f \circ U}(f(w)|v) + c_{f \circ U}$. □

Bemerkung 5.3 Aufgrund des obigen Satzes und der obigen Folgerungen fixieren wir im folgenden ein U, bzw. schreiben statt $K_U(w|v)$ einfach $K(w|v)$. Im Falle $v = e$ wird auch die Schreibweise $K(w) = K(w|e)$ verwendet.

Satz 5.5 *Sei $M \subseteq X^* \times X^*$, M rekursiv aufzählbar. Setze $M_v = \{w|(w,v) \in M\}$. Dann gibt es eine Konstante C mit $K(w|v) \leq |\log_q(|M_v|)| + C, q = |X| \forall w \in M_v$.*

Beweis. $\varphi'(\pi, v) = \pi$tes Paar der Form (w,v) in der Aufzählung von M ohne Berücksichtigung eventueller Wiederholungen. Da M rekursiv aufzählbar, ist es auch die Fkt φ'. Setze $\varphi(\pi, v) = pr_1(\varphi'(\pi, v))$, d.h. Projektion auf das erste Argument des Paares $\varphi'(\pi, v)$. Ist $w \in M_v \to \exists \pi \in X^*$ mit $\varphi(\pi, v) = w$ und $Q(\pi) \leq |M_v|$, d.h. $|\pi| \leq \log_q(|M_v|) + c$. Somit $K(w|v) \leq \log_q(|M_v|) + c_\varphi$. □

Folgerung 5.6

$$\frac{|\{w \mid K(w|v) \leq |\log_q(|M_v|)| - m\}|}{|M_v|} < \frac{q^{-(m-1)}}{q-1} q = |X|$$

Beweis. Die Anzahl verschiedener Programme π mit Länge $\leq k$ in X^* beträgt $\frac{q^{k+1}-1}{q-1} < \frac{q^{k+1}}{q-1}$ (geometrische Reihe), d.h. $|\{w|K(w|v) \leq k\}| < \frac{q^{k+1}}{q-1}$, mit $k = \lfloor \log_q(M_v) \rfloor - m \rightarrow q^k \leq |M_v| * q^{-m}$. □

Folgerung 5.7 *(1)* $\lim_{|w|\to\infty} K(w|v) = \infty$

(2) $\forall n \in \mathbb{N}, v \in X^*$ *gibt es ein* $w \in X^*$ *mit* $|w| = n$ *und* $K(w|v) \leq n$

Beweis. Verwendung des selben Abzählarguments wie im Beweis zur letzten Folgerung. □

Satz 5.8 *(1) Für kein* $v \in X^*$ *stimmt die Fkt* $K(\cdot, v) : X^* \to \mathbb{N}$ *mit einer partiell rekursiven Fkt.* $\varphi : X^* \to \mathbb{N}$ *mit unendlichem Definitionsbereich* $dom(\varphi)$ *auf* $dom(\varphi)$ *überein.*

(2) Es gibt eine rekursive Fkt $h : \mathbb{N} \times X^* \times X^* \to \mathbb{N}$ *die im ersten Argument monoton fallend ist, derart daß* $\lim_{t\to\infty} h(t, w, v) = K(w|v)$. *Es gilt somit u.a.* $K(w|v) \leq h(w, v, t) \forall t$.

Beweis.

(1) φ partiell rekursiv mit unendlichem Definitionsbereich $dom(\varphi)$, dann existiert $M \subseteq dom(\varphi)$ mit M ist rekursiv entscheidbar, M ist unendlich. Fixierung von v. Angenommen $K(w|v) = \varphi(w) \quad w \in dom(\varphi)$. Definiere $\chi : X^* \times X^* \to X^*$ durch :

$$\chi(\pi, u) = Q - \min\{w | w \in M, K(w|v) \geq Q(\pi)\}$$

wobei $Q(\pi)$ Nummer von π in der quasilexikographischen. Ordnung Q ist, und $Q - min(M)$ das Minimum der Menge M bzgl. Q ist. Da $K(w|v) = \varphi(w)$ $w \in dom(\varphi)$, ist χ ebenfalls partiell rekursiv. Da $\lim_{|w|\to\infty} K(w|v) = \infty$ ist $\chi(\cdot, v)$ immer definiert, d.h. χ ist eine rekursive Fkt. bei **fixiertem** v. Nach Definition gilt somit $K(\chi(\pi, v')|v) \geq Q(\pi) \forall\ \pi$. Andererseits $K(\chi(\pi, v)|v') \leq |\pi| + c_\chi$, d.h. $\forall \pi\ \exists c$: $Q(\pi) \leq |\pi| + c \leq \log(Q(\pi)) + c + 1$. Dies kann jedoch nicht sein! - Annahme falsch.

(2) Wissen $K(w|v) \leq |w| + c$, definiere $h : \mathbb{N} \times X^* \times X^* \to \mathbb{N}$ durch

$$h(t, w, v) = min(\{|w| + c\} \cup \{\pi : U(\pi, v) = w \quad \text{in nicht mehr als t-Schritten}\})$$

Wie man leicht sieht, um diese Floskel einmal zu verwenden, erfüllt h die Eigenschaften des Satzes.

□

Folgerung 5.9 *Es sei* $f : N \to \mathbb{N}$ *eine unbeschränkte rekursive Funktion. Dann ist* $A_v = \{w|K(w|v) \leq f(|w|)\}$ *rekursiv aufzählbar, während* $X^* \backslash A_v$ *keine unendliche rekursiv aufßählbare Teilmenge enthält, d.h. m.d.W. der Rekursionstheorie* A_v *ist einfach.*

Beweis.

- Mit obigem Satz gilt $A_v = \{w|\exists t \quad h(t,w,v) \leq f(|w|)\}$, berechne für alle Paare (w,t) $h(t,w,v), f(|w|)$, falls $h(t,w,v) \leq f(|w|)$ so zähle w auf.
- Angenommen es gibt $D \subseteq X^* \setminus A_v$, D unendlich und rekursiv aufzählbar. Dann existiert eine unendliche rekursiv entscheidbare Teilmenge $D' \subseteq D$. Definiere $g : X^* \times X^* \to X^*$ durch $g(\pi, v) = Q - \min\{w|w \in D', f(|w|) > Q(\pi)\}$. Zu mindestens einem $w \in D'$ existiert ein Wort w' mit $g(w', v) = w$, dann gilt $K(w|v) = K(g(w', v)|v) \leq |w'| + c_g \leq Q(w') + c_g$ und nach Definition von g gilt $Q(w') < f(|w|)$. Andererseits $w \in D' \to w \notin A_v \to f(|w|) < K(w|v)$. Wiederspruch, da natürlich $c_g \geq 0$.

□

Bemerkung 5.4 Im folgenden wird die Kolmogorov Komplexität als zahlentheoretische Fkt. aufgefaßt. $K : \mathbb{N} \to \mathbb{N}$ mit $K(n) = K(Q^{-1}(n))$, wobei $Q^{-1}(n)$ für das n-te Wort in X^* in der quasilexikographischen Ordnung steht.

Folgerung 5.10 *(1) $\exists c$, so daß $\forall n \in \mathbb{N}|K(n+1) - K(n)| < c$ gilt.*

(2) $\exists c$, so daß $\forall w \in X, \forall x \in X^*|K(wx) - K(x)| \leq c$

(3) $\exists c$, so daß $\forall n \in N \quad K(n) \leq \log_q(n) + c(q > 1)$ gilt.

(4) $\lim_{n\to\infty} K(n) = \infty$

(1),(2),(3),(4) sind einfache Folgerungen bereits bekannter Eigenschaften der Kolmogorov Komplexität von Wörtern.

Bemerkung 5.5 Definiere $\kappa : \mathbb{N} \to \mathbb{N}$ durch $\kappa(n) = min\{K(m)|m \geq n\}$, d.h. $\kappa(m)$ ist eine untere Schranke für die Kolmogorov Komplexität aller natürlichen Zahlen m mit $m \geq n$, aus (4) der letzten Folgerung folgt sofort $\lim_{n\to\infty} \kappa(n) = \infty$.

Satz 5.11 *Es sei $\varphi : \mathbb{N} \to \mathbb{N}$ eine beliebige partiell rekursive Fkt, die auf ihrem Definitionsbereich monoton wachsend ist. Dann $\exists n_0 \in N$ mit $\kappa(n) < \varphi(n) \quad \forall n \in dom(\varphi), n \geq n_0$*

Beweis. Angenommen es gibt $M \subseteq dom(\varphi)$, M unendlich und rekursiv, mit $\varphi(n) \leq \kappa(n) \quad \forall n \in M$, o.B.d.A. gelte $0 \in M$. Definiere $\psi : \mathbb{N} \to \mathbb{N}$ durch

$$\psi(n) = \begin{cases} max\{\varphi(n) - 1, 0\} & : \quad n \in M \\ \psi(max\{j|j \leq n, j \in M\}) & : \quad n \notin M \end{cases}$$

Die Funktion ψ ist monoton wachsend, und für jede natürliche Zahl definiert, d.h. rekursiv. Definiere die Hilfsfunktion. $\chi : \mathbb{N} \to \mathbb{N}$ durch $\chi(l) = max\{n|\kappa(n) \leq l\}$ (*) Weiter definiere die Hilfsfunktion. $F : \mathbb{N} \to \mathbb{N}$ durch $F(l) = max\{n|\psi(n) \leq l\}$, F ist eine rekursive Fkt.. Mit (*) folgt sofort $\chi(l) + 1 = min\{n|\kappa(n) > l\}$ (**), d.h. jede Zahl größer als $\chi(l)$ hat eine Kolmogorov Komplexität größer als l (***). Betrachte nun die Menge

$$\omega = \{l : min\{n|\kappa(n) > l\} \leq max\{n|\psi(n) \leq l\}\}$$

bzw. die Definition etwas anders formuliert $\omega = \{l : \chi(l) + 1 \leq F(l)\}$. Die Menge ω ist unendlich, da unendlich oft $\psi(n) < \varphi(n) \leq \kappa(n)$. Aus (*) folgt $\chi(\psi(n)) < min\{n|k(n) >$

$\psi(n)\}$, d.h. $\psi(n) < \kappa(n) \rightarrow \psi(n) \in \omega$, ebenso gilt $\chi(l) < F(l)$ bzw. mit (***) $K(F(l)) > l \quad \forall l \in \omega$. Es gilt jedoch $K(F(l)) \leq \log_q(l) + c_F$, da F rekursiv, d.h. $l < K(F(l)) \leq \log_q(l) + C_F q > 1 \quad \forall l \in N$. Dies ist jedoch unser sehnlichst gewünschter Widerspruch, da ω eine unendliche Menge, d.h. Annahme falsch, Satz richtig. □

Bemerkung 5.6 Hoffentlich liest in näherer Zukunft mal jemand diesen Beweis, die Arbeit an diesem Beweis war etwa so, wie ich mir das Melken von Mäusen vorstelle.

5.1 Anwendungen der Kolmogorov Komplexität

1970 bewies Wolfgang Paul eine bis dato unbekannte untere Schranke für die Komplexität eines Sortierverfahrens, ausgehend von dieser Arbeit, stellte sich heraus, daß die Kolmogorov Komplexität Abschätzungen für die mittlere Komplexität von Berechnungsprozessen liefern konnte. Des weiteren kann der folgende Satz mittels Kolmogorov Komplexität bewiesen werden.

Satz 5.12 *Es gibt eine Familie von regulären Sprachen, die mit einem NDEA Automaten mit $n+1$ Zuständen entschieden werden kann, aber durch keine DEA mit weniger als $2^{O(n)}$ Zuständen akzeptiert werden.*

Satz 5.13 *Es sei M eine einbändige Turingmaschine ohne Eingabeband, die die Sprache*

$$W = \{wc^{|w|}w : w \in \{a, b\}\}$$

entscheidet, dann gilt $F_M(3 * n) \geq c * n^2$, wobei

$$F_M(l) = \frac{\sum_{u \in W, |u|=l} t_M(u)}{|\{u | u \in W, |u| = 1\}|}$$

die mittlere Zeit der Akzeptierung eines Wortes der Länge l ist.

Bemerkung 5.7 Die obigen Ergebnisse können mit Blick auf die Problematik der Datenkompression wie folgt interpretiert werden. Es gibt einen universellen Algorithmus U zur Datendekompression, der bis auf eine endliche Konstante für alle Texte optimal ist.

$$\forall \varphi \quad K_U(w|v) \leq K_\varphi(w|v| + c_\varphi)$$

Der Nachteil dieser unteren Schranke ist jedoch ihre geringe Quantitativität und vermutlich die Tatsache $c_\varphi \gg 1$, m.a.W. für spezielle Wahl von φ ist trotz allem für Texte hoher Komplexität eine starke Kompression möglich. Des weiteren gibt es keinen zu U gehörenden universellen Kompressionsalgorithmus.
Ein großer Anteil der Texte w besitzt eine Komplexität $\geq |w| - m$ und nur ein kleiner Teil wird echt komprimiert, setze $M_v = X^{Q(v)}$, $M = \{(w, Q^{-1}(|w|))|w \in X^*\}$ und benutze Folgerung 5.6.

6 Kolmogorov Komplexität unendlicher Folgen

Wie gehabt X Alphabet, $|X| = q \geq 2$, Sei ξ eine unendliche Folge der Form $x_1 \dots x_i \dots \quad i \in \mathbb{N} \setminus \{0\}$, definiere $K(\xi|\cdot) : \mathbb{N} \to \mathbb{N}$ durch $K(\xi|n) =$ Komplexität des Präfixes von ξ der Länge n. Definiere $K'(w|n) : X^* \to \mathbb{N}$ durch

$$K'(w|n) = \begin{cases} K(w|Q^{-1}(n)) & : \quad n = |w| \\ n.d. & : \quad n \neq |w| \end{cases}$$

Beachte den folgenden Unterschied :

- $M_1 = \{w|K(w) \leq c\}$ ist endliche Menge für jede Konstante c
- $M_2 = \{w|K'(w||w|) \leq c\}$ ist eine unendliche Menge, falls c hinreichend große Konstante, da dann $\{0^n|n \in N\}$ eine Teilmenge von M_2 ist.

Im folgenden wird wiederum nur aus reiner Faulheit $K(w||w|)$ statt $K'(w||w|)$ geschrieben.

Folgerung 6.1

$$\forall n, m \in \mathbb{N} \quad |\{w|K(w|n) \leq m\}| < \frac{q^{m+1}}{q-1}$$

$$\forall n, m\,(\geq 6) \in \mathbb{N} \quad |\{w|K(w|n) \leq n-m\}| < q^n - \frac{q^{n-m}}{q-1} > 0$$

Beweis. Siehe Beweis zu Folgerung 5.6 □

Satz 6.2 (Satz von Peter Martin-Löf (1)) *Für jede Folge ξ über X gibt es eine Konstante c, so daß*

$$K(\xi|n) \leq n - \log_q(n) + c$$

für unendlich viele $n \in \mathbb{N}$ gilt, m.a.W. es existiert keine Konstante c mit $|K(\xi|n) - n)| \leq c$.

Beweis. In geeigneten Fällen enthält n mehr Informationen über den zu erzeugenden Text, als nur die Länge des Textes. Definiere $\varphi : X^* \times X^* \to X^*$ durch

$$\varphi(\pi, n) = \begin{cases} Q^{-1}(|\pi|)\pi & : \quad |Q^{-1}(\pi)\pi| = n \\ n.d. & : \quad |Q^{-1}(\pi)\pi| \neq n \end{cases}$$

Mithin $|\varphi(\pi, n)| = n \simeq |\pi| + \log_q(|\pi|)$, bzw. $K'(\varphi(\pi, n)|n) = K'(Q^{-1}(|\pi|)\pi|n) \leq |\pi| + c_1$ $= |Q^{-1}(\pi|)\pi| - |Q^{-1}(\varphi)| \leq n - \log_q(n) + c_2$, dies natürlich nur unter der Voraussetzung $|Q^{-1}(|\pi|)\pi| = n$. Der Beweis ist nun erbracht, da jede unendliche Folge ξ über X unendlich viele Präfixe der Form $Q^{-1}(|\pi|)\pi$ besitzt, da für jedes Präfix $v \sqsubseteq \xi$ das Präfix w der Länge $|w| = |v| + Q(v)$ die gewünschte Gestalt besitzt, mithin $K(w||w|) \leq |w| - \log_q(|w|) + c$ für unendliche Präfixe von ξ. □

Bemerkung 6.1 Im Beweis wurde die Hilfsaussage $\log_q(m) \geq \log_q(\frac{m+\log_q(m)}{m}) = \log_q(m + \log_q(m)) - 1$ verwendet.

Satz 6.3 (Satz von P. Martin-Löf (2)) *Es gibt 'viele' Wörter mit $K(w||w|) \geq |w| - \log_q(|w|) + c$.*

Beweis. Definiere $\varphi :\to X^* \times N \to X^*$ durch $\varphi(\pi, n) = Q^{-1}(n)0^{n-|Q^{-1}(n)|}$, $Q^{-1}(n)$ kann dabei als Darstellung der Zahl n aufgefaßt werden und der Bestandteil $0^{n-|Q^{-1}(n)|}$ kann als führender Nullterm interpretiert werden. Es gilt $|\varphi(\pi, n)| = n$ und $K(\varphi(\pi, n)|n) \leq c$, da bei Kenntnis von n natürlich $\varphi(\pi, n)$ reproduziert werden kann. Verallgemeinernd betrachte eine Folge natürlicher Zahlen $n_1, n_2, n_3 \ldots$, Worte w_i der Form $w_1 = Q^{-1}(n_1)0^{f(n_1)}$, $w_2 = w_1 Q^{-1}(n_2)0^{f(n_1,n_2)}, w_3 = w_2 Q(n_3)0^{f(n_1,n_2,n_3)} \ldots$ mit $f(n_1, \ldots, n_i) = n_i' - |Q^{-1}(n_i')|$ und $n_1' = Q(Q^{-1}(n_1))$, m.a.W. $(n_1' = n_1)$, $n_2' = Q(w_1 Q^{-1}(n_2))$, $n_3' = Q(w_2 Q^{-1}(n_3))$, $\ldots$, d.h. für jedes der Wörter w_i gibt es ein n_i' mit $\varphi(\pi, n_i') = w_i$, d.h. $K(w_i|n_i') \leq c$. In irgendeiner Übungsaufgabe wird $K(w \cdot w'||w \cdot w'|) \leq K(w||w|) + |w'| + 2 * \log_q(|w'|) + c$ bewiesen, d.h. (*) $K(w||w|) \leq K(w \cdot w'||w \cdot w'|) - |w'| - 2 * \log_q(|w'|) - c$, ebenfalls läßt sich (**) $K(w'||w'|) - |w| - 2 * \log_q(|w|) \leq K(w \cdot w'||w \cdot w'|)$ beweisen. Wähle die n_k derart, daß $K(Q^{-1}(n_k)||Q^{-1}(n_k)|) \geq |Q^{-1}(n_k)| - 1$, wobei insbesondere über $l_k = |Q^{-1}(n_k)|$ noch frei zu verfügen ist. Die Komplexität $\gamma_i = K(Q^{-1}(n_i')||Q^{-1}(n_i')|) = K(w_{i-1}Q^{-1}(n_i)||w_{i-1}Q^{-1}(n_i)|)$ läßt sich nun wie folgt abschätzen : (**) liefert nun $\gamma_i \geq l_i - 1 - |w_{i-1}| - 2 * \log_q(|w_i|) - c^* \geq l_i - 3 * |w_i| - c^{**}$. Wähle nun l_i derart, daß $\gamma_i \geq n - |n| + c$ mit $n = Q^{-1}(n_i')$. □

Satz 6.4 (Satz von P. Martin-Löf (3)) *Ist $f : \mathbb{N} \to \mathbb{N}$ eine Funktion, für die*

$$\sum_{n \in \mathbb{N}} q^{-f(n)} < \infty$$

gilt, so gibt es eine Folge ξ über X für die $K(\xi|n) \geq n - f(n) + c$ fast überall gilt.

Beweis. Wähle k derart, daß $\sum_{n=k}^{\infty} q^{-f(n)} < \frac{1}{q}$, betrachte nun alle Wörter w mit Länge $n \geq k$ für die ein $i \in [k, n]$ existiert, so daß für das Präfix w_i die Abschätzung $K(w_i|i) \leq i - f(i)$ erfüllt ist. Ein solches w_i existiert, da

$$\Omega = \{w||w| = n, \exists v \quad v \sqsubseteq w, K(v||v|) \leq |v| - f(|v|)\} \subseteq M = \bigcup_{k \leq i \leq n} M_i \cdot X^{n-|v|}$$

mit M_i definiert durch

$$M_i = \{v||v| = i, K(v||v|) \leq |v| - f(|v|)\}$$

Nach einer am Anfang in diesem Kapitel stehenden Folgerung gilt

$$|M_i| \leq \frac{q^{|v|-f(|v|)+1}}{q-1}$$

d.h.

$$|M| \leq \sum_{i=k}^{n} \frac{q^{v|-f(|v|)+1}}{q-1} * q^{n-|v|} = \frac{q^{n+1}}{q-1} * \sum_{i=k}^{n} q^{-f(i)} < q^n$$

m.a.W wir haben nun die Existenz eines Wortes $\omega \in X^n \setminus \Omega$ mit

- $|\omega| = n$
- $\forall v \sqsubseteq \omega, |v| \geq k \to K(v||v|) > |v| - f(|v|)$

□

Bemerkung 6.2 Der 2. Satz von P. Martin-Löf ist kein Spezialfall des 3., da für f mit $f(n) = \log_q(n)$ nicht die Voraussetzungen des 3. Satz erfüllt ist, die harmonische Reihe $\sum_{n\in\mathbb{N}} \frac{1}{n} = \sum_{n\in\mathbb{N}} q^{-f(n)}$ ist divergent.

Lemma 6.5 (Königsches Unendlichkeits Lemma von Denés König) Es sei $G = (V, E)$ ein (unendlicher) gerichteter Graph, für den jeder Knoten einen endlichen Grad hat. Gibt es in G einen Pfade unbeschränkter Länge, so gibt es auch einen unendlichen Pfad in G.

Bemerkung 6.3 Die Endlichkeit des Grades eines jeden Knoten ist wichtig, daß Lemma ist falsch, falls diese Voraussetzung verletzt ist. z.B. Betrachte man einen Baum, von der Wurzel des Baumes gehe für jede natürliche Zahl n ein Pfad mit Länge $n-1$ in die Tiefe. Dieser gerade beschriebene Graph hat Pfade unbeschränkter Länge, aber keinen unendlichen Pfad.

Definition 6.1 Eine Folge ξ über X heißt rekursiv $\iff$ ξ als Abbildung von $\mathbb{N} \to X$ ist rekursiv.

Satz 6.6 *ξ rekursive Folge* $\iff \exists c \quad K(\xi|n) \leq c.$

Beweis.

- Da ξ rekursiv, gibt es eine rekursive Fkt. $\varphi : X^* \times \mathbb{N} \to X$ mit $\varphi(\pi, n) = \xi(n)$, d.h. kennt man n berechnet man $\varphi(\pi, 1), \varphi(\pi, 2) \ldots \varphi(\pi, n)$, und gibt $\varphi(\pi, 1), \varphi(\pi, 2) \ldots$ aus, da diese Berechnung unabhängig von π ist, folgt $K(\xi|n) \leq c$. Dies ist im grundegenommen nur eine Folgerung der Tatsache $K(\varphi(e, v)|v) \leq c$ für beliebiges v und eine beliebige rekursive Fkt φ.

- Fixiert man Konstante c und betrachtet die Menge $M = \{\xi | \forall n \quad K(\xi|n) \leq c\}$, so stellt man fest, daß $|M| \leq q^{c+1}$, d.h. M eine endliche Menge ist. Mit der Gesamtheit der Programme der Länge $\leq c$ können nicht mehr als q^{c+1} Ausgaben der Länge n produziert werden, d.h. betrachtet man alle vorkommenden w mit $|w| = n$, $\exists \xi \in M \sqsubseteq \xi$, so ist die Anzahl der $w \leq q^{c+1}$, n ist hierbei jedoch beliebig. $M = \{\xi_1, \ldots, \xi_b\}$. Da $\xi_i \neq \xi_j \quad i \neq j$, folgt die Existenz von Wörtern w_i mit $w_i \sqsubseteq \xi_i, w_i \not\sqsubseteq \xi_j \quad i \neq j$, m.a.W. w_i identifiziert eindeutig ξ_i. Da M endlich gibt es l_c mit $|w_i| \leq l_c$, o.B.d.A gelte $|w_i| = l_c \forall i$. Falls w, lc gegeben, so kann man $\xi(n)$ folgenderweise berechnen

  - $n < l_c$, dieser Fall ist trivial, oder wie der praktische Informatiker sagt : mittels eines **table-look-up**, was natürlich besser klingt. zu erledigen.
  - $n \geq l_c$, Probiere im Dovetail-Verfahren alle Programme der Länge $\leq c$ als Eingabe von $U(\pi, n)$ aus, falls $w \sqsubseteq U(\pi, n)$, so $u(\pi, n) \sqsubseteq \xi$ nach Konstruktion, und da $|U(\pi, n)| = n$ hat man $\xi(n)$.

□

7 EA auf unendlichen Wörtern

Bezeichne mit X^ω die Menge aller unendlichen Worte über X (des Ordnungstyps ω). $\xi \in X^\omega \to \xi = x_1 x_2 \ldots \quad x_i \in X$, von dieser Definition ausgehend definiert man W^ω $w \subseteq X^*$ durch $\xi \in W^\omega \to \xi = w_1 w_2 \ldots \quad w_i \in W$, man nennt W^ω eine ω Sprache.. Sei $A = (X, Z, z_0, \delta)$ ein im herkömmlichen Sinne definierter EA, der einzige Unterschied im folgenden ist, daß die Eingabe ξ ein Element aus X^ω ist. Als vollständigen Lauf (run) von A auf $\xi \in X^\omega$ mit $\xi = x_1 x_2 \ldots$ bezeichne die Sequenz

$$z_0 \binom{x_1}{z_1} \binom{x_2}{z_2} \ldots \binom{x_i}{z_i} \ldots$$

mit $(z_0, x_1, z_1), (z_{i-1}, x_i, z_i) \in \delta$. Die Sequenz $\eta = z_0 z_1 \ldots z_{i-1} z_i z_{i+1} \ldots$ bezeichne als Lauf von A auf ξ, beachte $\eta \in Z^\omega$. Versuch der Charakterisierung von Teilmengen von X^ω mit Hilfe der endlichen Automaten. Da die Eingabe ein Wort aus X^ω ist, hält der Automat zu keinem endlichen Zeitpunkt., d.h. die Akzeptierungsbedingung muß modifiziert werden.

Bemerkung 7.1 Es wird die folgende Schreibweise vereinbart : $\inf(\eta)$ sei die Menge aller in $\eta \in Z^\omega$ unendlich oft vorkommenden Zustände.

Definition 7.1 (Akzeptierungsbedingungen) Es sei $A = (X, Z, z_0, \delta)$ ein NEA, $M \subseteq X^\omega$, $Z' \subseteq Z, T \subseteq 2^Z \times 2^Z, F \subseteq 2^Z$.

(a) A akzeptiert M mit der Büchi Bedingung, falls

$$M = \{\xi | \text{es gibt Lauf } \eta \text{ auf } \xi \text{ mit } \inf(\eta) \cap Z' \neq \emptyset\}$$

(b) A akzeptiert M mit der Rabin Bedingung, falls

$$M = \{\xi | \text{es gibt Lauf } \eta \text{ auf } \xi, (P, Q) \in T \text{ mit } \inf(\eta) \cap P = \emptyset, \inf(\eta) \cap Q \neq \emptyset\}$$

(c) A akzeptiert M mit der Muller Bedingung, falls

$$M = \{\xi | \text{es gibt Lauf } \eta \text{ auf } \xi, Q \in F \text{ mit } \inf(\eta) = Q\}$$

Bemerkung 7.2 Wie man leicht sieht ist die Muller Bedingung die schärfste Bedingung und die Büchi Bedingung die schwächste.

Bemerkung 7.3 In der Vorlesung wurden drei wesentlich verschiedene DEA angegeben mit zwei Zuständen, die alle nur X^ω-Wörter mit unendlich vielen b akzeptieren, verwendet wurde die Muller Bedingung, d.h. minimale DEA müssen nicht eindeutig sein.

Satz 7.1 *Die Menge der von NEA bzgl. der Büchi-, Rabin- bzw. Muller Bedingung akzeptierten ω-Sprachen stimmt über ein.*

Beweis. Man kann die Büchi-Bedingung als Rabin-Bedingung formulieren und die Rabin-Bedingung als Muller-Bedingung. Dies wurde als freiwillige Übungsaufgabe in der Vorlesung gestellt. Um den Beweis zu erbringen, konstruieren wir nun aus einem Muller-Automaten $\mathcal{M}$ einen Büchi-Automaten B. Sei $\mathcal{M} = (X, Z, z_0, \delta)$ und $\mathcal{T} = \{Z_1, \ldots, Z_n\}$, setze

$$\mathcal{B} = (X, Z \cup (Z \times M \times \{1, 0\}), z_0, \delta')$$

mit $M = \{(i,T)\,|1 \leq i \leq n, T \subseteq Z_i\}$ und $Z'_{\mathcal{B}} = Z \times M \times \{1\}$.

$$\delta'_0 = \delta$$

Unser zu konstruierender Automat soll genau wie der Automat $\mathcal{M}$ arbeiten.

$$\delta'_1 = \{(Z, x, (z', (i, \{z'\}), 0)) \,|(z, x, z') \in \delta, z' \in Z_i\}$$

In diesem Bestandteil wird der Zeitpunkt. t^*, an dem der Automat die Menge Z_i nicht mehr verläßt, geraten.

$$\delta'_2 = \{((z, (i,T), 0), x, (z', (i,T'), 0)) \,|\, (z, x, z') \in \delta, T' = T \cup \{z'\} \subset Z_i\}$$

In diesem Bestandteil wird die Menge Z_i in T aufgesammelt, solange $T \neq Z_i$, in diesem Falle steht $\subset$ für echte Teilmenge.

$$\delta'_3 = \{((z, (i,T), 0), x, (z', (i, Z_i), 1)) \,|\, (z, x, z') \in \delta, T \cup \{z'\} = Z_i\}$$

Die geratene Menge ist einmal vollständig durchlaufen worden.

$$\delta'_4 = \{((z, (i, Z_i), 1), x, (z', (i, \{z'\}), 0)) \,|\, (z, x, z') \in \delta, z' \in Z_i\}$$

Start eines weiteren Versuchs die Menge Z_i vollständig zu durchlaufen. Falls wir t^* richtig geraten haben bleibt der Automat dauerhaft in $Z_i \in \mathcal{T}$.

$$\delta' = \delta'_0 \cup \delta'_1 \cup \delta'_2 \cup \delta'_3 \cup \delta'_4$$

□

Satz 7.2 (Satz von Büchi) *Eine ω-Sprache F wird von einem NEA akzeptiert $\iff$ es gibt ein $n \in \mathbb{N}$, reguläre Sprachen W_i, V_i, mit $F = \bigcup_{i=1}^{n} W_i V_i^{\omega}$, d.h. $\xi \in F \to \xi = wv_1v_2\ldots$, m.a.W. periodische Struktur mit einer Vorperiode.*

Beweis.

- $A = (X, Z, z_0, \delta)), Z' \subseteq Z$ akzeptiere F, d.h. $\forall \xi \in F$ gibt es einen Lauf η mit $z' \in \inf(\eta) \cap Z' \neq \emptyset$, Büchi Bedingung, d.h. $\xi = wv_1v_2\ldots$, und

  $$z_0 \xrightarrow{w} z' \xrightarrow{v_1} z' \xrightarrow{v_2} \ldots$$

  Hierbei steht $z \xrightarrow{u} z'$ für einen Pfad im Automaten A der von z nach z' führt und mit u markiert ist. Sei $n = |Z'|$, d.h $Z' = \{z_1, \ldots z_n\}$, so setze $W_i = \{w | z_0 \xrightarrow{w} z_i\}$ und $V_i = \{v | z_i \xrightarrow{w} z_i\}$, z.Z. ist nun, daß die von A bzgl Z' akzeptierte Menge mit $\cup_{i=1}^{n} W_i V_i^{\omega}$ übereinstimmt.

  (a) ξ werde von A bzgl. Z' akzeptiert, d.h. es gibt Lauf η auf ξ mit $\inf(\eta) \cap Z' \neq \emptyset$. Sei $z_i \in \inf(\eta) \cap Z'$, dann gibt es Zeitpunkt. $t_j, t_j < t_{j+1}, j = 0, 1, 2\ldots$ mit $\eta(t_j) = z_i$. Setze $\xi\,(t_j, t_{j+1}] = \xi(t_j + 1)\ldots\xi(t_{j+1})$, dann gilt nach Definition $\xi\,(0, t_1] \in W_i$ und $\xi\,(t_j, t_{j+1}] \in V_i$, d.h. $\xi \in W_i V_i^{\omega}$

  (b) $\xi \in W_i V_i^{\omega}$, dann wird ξ von A bzgl $\{z_i\} \subseteq Z'$ akzeptiert, und es gibt einen unendlichen Lauf von A auf ξ der unendlich oft durch z_i verläuft.

- Es gilt (1) $e \in W \rightarrow WV^\omega = V^\omega \cup (W \setminus \{e\})$ und (2) $V^\omega = VV^\omega = (V \setminus e)^\omega$, d.h. o.B.d.A. kann angenommen werden $e \notin W_i, e \in V_i$. Da W_i, V_i reguläre Sprachen sind, gibt es NEA A_i die W_i, und NEA B_i die V_i akzeptieren, Konstruiere nun NEA C_i durch von A_i und B_i und legt jeweils den Endzustand von A_i mit dem Anfangs - bzw. Endzustand von B_i zusammen, nenne diesen Zustand c_i^*. Anfangszustand von C_i wird der verwende den von A_i. C_i akzeptiert mit $\{c_i^*\}$ die Menge $W_iV_i^\omega$. Legt man nun die Anfangszustände der C_i zusammen, so gewinnt man einen NEA C, der mit $\{c_i^* | 1 \leq i \leq n\}$ die Menge F mit der Büchi Bedingung akzeptiert.

□

Bemerkung 7.4 In Anlehnung an die Theorie endlicher Wörter, bezeichne ω-Sprachen, die von NEA entscheidbar sind, d.h. von der im Satz von Büchi angegebenen Gestalt, als reguläre ω-Sprachen.

Bemerkung 7.5 In der Theorie endlicher Wörter sind NEA und DEA gleichwertig. Betrachtet man jedoch $F = \{a,b\}^*a^\omega$, so stellt man fest, daß es keinen DEA $A = (X, Z, z_0, \delta), Z' \subseteq Z$ gibt, der F mit der Büchi-Bedingung akzeptiert. Angenommen Behauptung ist falsch, da A DEA kann es für $\xi \in F$ nur einen akzeptierenden Lauf $\eta(\xi)$ auf A geben. Es gilt nun jedoch $\xi \in F \iff \inf(\eta(\xi)) \cap Z' \neq \emptyset$. Da $a^\omega \in F$ gibt es $z_1' \in \inf(\eta(a^\omega)) \cap Z'$. Sei $w_1 \sqsubset a^\omega$ und $\delta(z_0, w_1) = z_1' \in Z'$. Da $w_1ba^\omega \in F$ gibt es $z_2' \in \inf(\eta(w_1ba^\omega)) \cap Z'$ und folglich gibt es w_2 mit $w_1b \sqsubseteq w_2 \sqsubset w_1ba^\omega$ und $\delta(z_0, w_2) = z_2'$. Induktiv bestimme nun die w_i nun so, daß $\delta(z_0, w_i) \in Z'$, das Wort w_i mindestens i mal den Buchstaben b enthält und $w_ib \sqsubseteq w_{i+1} \sqsubset w_iba^\omega$ für jedes i gilt. Im Grenzprozess mit $i \rightarrow \infty$ erhält man somit ein $\xi \in (a^*b)^\omega$. Es gilt $(a^*b)^\omega \cap \{a,b\}a^\omega = \emptyset$, d.h. $\xi \notin F$, aber $\inf(\eta(\xi)) \cap Z' \neq \emptyset$, d.h. Annahme der Existenz eines DEA ist falsch. Folglich sind DEA und NEA bei Verwendung der Büchi-Akzeptierung nicht äquivalent.

Bemerkung 7.6 Im folgenden soll nun versucht werden mit Hilfe der Nerode Kongruenz, das Verhältnis von DEA und NEA zu charakterisieren.

$$w \simeq_F v \quad w, v \in X^* \quad \iff \quad \forall \xi \in F \quad w\xi \in F \iff v\xi \in F$$

Lemma 7.3 Ist F reguläre ω-Sprache, so hat $\simeq_F$ endlichen Index.

Beweis. F werde vom NEA $A = (x, Z, z_0, \delta), Z' \subseteq Z$ mit Büchi-Bedingung akzeptiert. Setze

$$\gamma(w) = \{z | z \in Z, z_0 \xrightarrow{w} z\}$$

und sage $w \equiv_A v$, falls $\gamma(w) = \gamma(v)$. Wie man leicht sieht, ist $\equiv_A$ eine Äquivalenzrelation mit endlichem Index, da Potenzmenge von Z endlich ist. Es gilt $w \equiv_A v \rightarrow w \simeq_F v$, da aus $w\xi \in F$ auch $v\xi \in F$ folgt, da nur ein akzeptierender Lauf benötigt wird, der die Büchi-Bedingung erfüllt und $\gamma(w) = \gamma(v)$. □

Bemerkung 7.7 Betrachte nun $F = \{a,b\}a^\omega$, d.h. Menge aller ω-Wörter mit endlich vielen b, es folgt nun

$$w \simeq_F v \quad w, v \in X^* \iff \forall \xi w\xi \in \{a,b\}^*a^\omega \iff v\xi \in \{a,b\}^*a^\omega$$

Diese Bedingung an Paare (w, v) ist jedoch immer erfüllt, d.h. Index der Nerodeschen Rechtskongruenz ist 1, $\simeq_F$ ist somit nicht zur Charakterisierung von F geeignet.

Bemerkung 7.8 Es gibt nicht reguläre ω-Sprachen F, für die $\equiv_F$ endlichen Index hat. F reguläre, so $F = \bigcup_{i=1}^{n} W_i V_i^\omega$ nach Satz von Büchi, d.h. es gibt w, v mit $wv^\omega \in F$. Definiere $Nult = \{\xi | \xi \text{ ist nicht von der Form } wv^\omega\}$, $Nult$ steht für non-ultimately periodic. $\simeq_{Nult}$ hat immer Index 1, da $w \simeq_{Nult} v$ immer gilt. Zur kurzen Charakterisierung : die regulären ω-Sprachen entsprechen in etwa den rationalen Zahlen, während die $Nult$ Menge der irrationalen Zahlen repräsentiert.

Bemerkung 7.9 Ein weiterer Ansatz zur Konstruktion eines DEA aus einem NEA, der aus der Theorie der endlichen Wörter kommt, ist die Potenzmengen Konstruktion, in ein NEA angegeben mit zwei Zuständen, der nur die leere Menge akzeptiert, der nach der Potenzmengen Konstruktion entstandene Automat akzeptierte X^ω. Untersucht man die Gründe für das Versagen der Potenzmengen Konstruktion, stellt man fest, daß für diese Konstruktion nur wichtig ist, wo komme ich hin, beim verschmelzen der Pfade kann es geschehen, das es unmöglich wird zu entscheiden, ob der Durchlauf eines Finalzustandes parallel oder hintereinander stattfand, d.h. Information geht verloren. Ein illustratives Beispiel wurde in der Vorlesung angegeben. Es ist jedoch möglich die Potenzmengen Konstruktion zu modifizieren und erfolgreich die Determinisierung von ω-Automaten durchzuführen, d.h. aus einem NEA Büchi einen DEA Rabin zu konstruieren. Die Konstruktion von SAFRA ist mal wieder ein Beispiel für die Richtigkeit des Satzes : 'Ordnung ist das halbe Leben.'

Konstruktion von SAFRA

(a) Simulation einer Zeitskala mittels Einführung sogenannter Wechselpunkt (breakpoints), die dann in die Akzeptierungsbedingung $\mathcal{T}$ der Rabin-Bedingung eingehen.

(b) Zusammenführung von Zuständen bei der Potenzmengen Konstruktion nur dann, wenn zwischenzeitlich, d.h. seit dem letzten Wechselpunkt. ein Finalzustand durchlaufen werden konnte, d.h. Identifikation von z_1 mit z_2 falls

$$z_1 \xrightarrow{z^1 \in Z} z, z_2 \xrightarrow{z^2 \in Z} z$$

jedoch keine Identifikation von z_1 mit z_3 falls

$$z_3 \xrightarrow{\nexists z^3 \in Z} z$$

Prinzip der Verzweigung und der Wiedervereinigung von Makrozuständen
Makrozustände sind die zum zu konstruierenden Automaten gehörenden Zustände, d.h. bei einer Verzweigung wird ein neuer Zustand eingeführt, und die Wiedervereinigung von Makrozuständen steht für die Identifikation zweier solcher Zustände.

$$R_0 \xrightarrow{w_{11}} P_1 \xrightarrow{w_{12}} R_1 \xrightarrow{w_{21}} P_2 \ldots$$

mit $R_0 \subseteq Z$, $\emptyset \neq Q_1 = P_1 \cap Z'$ (Verzweigungspunkt.), an dieser Stelle wird der neue Zustand Q_1 eingeführt. Sollte man ausgehend von Q_1 mittels Potenzmengenbildung nach Konsumierung von w_{12} eine Menge Q_1' erreichen, wobei $R_1 = Q_1'$ mit Freuden und Entzücken festgestellt werden kann, so wurde ein Wechselpunkt erreicht, und die Pfade $R_0 \to P_1 \to R_1$ bzw. $R_0 \to Q_1 \to Q_1'$ können identifiziert werden, m.a.W. man

kann $Q_1' = R_1$ setzen. Das ganze wird nun mit R_1 wiederholt. Der von Q_1 ausgehende Pfad befindet sich auf 'Ebene' $i+1$, falls der Pfad $R_0 \to P_1 \to R_1$ sich auf der 'Ebene' i befand. Kommt man unendlich oft durch Wechselpunkt. und Wiedervereinigungspunkt, so durchläuft man unendlich oft Finalzustände, d.h. man erfüllt die Büchi-Akzeptierungsbedingung. Algebraisiert lautet dies wie folgt:

$$\forall z, z' \text{mit} \quad z \in R_i, z' \in R_{i+1} \text{und} \quad z \xrightarrow{w_1} z' \text{folgt} \quad \exists z'' \in Z' \; z \to \overset{w_1}{z''} \to z'$$

Lemma 7.4 (Wiedervereinigungslemma) Es sei $R_0 = \{z_0\}$ und weiter seien P_i, Q_i, Q_i', R_i eine Folge von Makrozuständen des Automaten A, wie oben definiert wurde. Dann gibt es einen Lauf η von A auf $\xi = w_1 w_2 \ldots$ mit $\inf(\eta) \cap Z' \neq \emptyset$.

Prinzip der Beseitigung von Redundanz
Seien $\mathcal{P}_1, \mathcal{P}_2$ Pfade auf der selben Ebene, wobei $\mathcal{P}_1$ früher als $\mathcal{P}_2$ abgezweigt wurde, so werden Zustände die zum gleichen Zeitpunkt. in beiden vorkommen aus $\mathcal{P}_2$ entfernt, ist $\mathcal{P}_2$ zu einem Zeitpunkt leer, so stirbt der Pfad ab, der spätere Pfad stirbt ab. Die Konstruktion liefert eine baumähnliche Struktur. Stopp der Konstruktion falls keine neuen Makrozustände mehr entstehen.
Datenstruktur SAFRA Baum

Satz 7.5 (Satz von McNaughton) *Es sei $A = (X, Z, z_0, f)$ ein NEA der mit $Z' \subseteq Z$ die ω-Sprache F mit der Büchi-Bedingung akzeptiert. Dann gibt es einen DEA $B = (X, S, s_0, g)$ und eine Tafel $\mathcal{T} \subseteq 2^Z \times 2^Z$ derart, daß B mit $\mathcal{T}$ die ω-Sprache F mit der Rabin-Bedingung akzeptiert.*

Bemerkung 7.10 Im folgenden bezeichne R_ω die Menge der regulären ω-Sprachen über dem Alphabet X, und es sei DR_ω die Menge der von DEA Büchi-akzeptierten ω-Sprachen über dem Alphabet X.

Folgerung 7.6 *R_ω ist abgeschlossen bzgl. Boolescher Operationen (Durchschnitt-, Vereinigung- bzw. Komplementbildung).*

Beweis. Angenommen A akzeptiert mit $\mathcal{F}$ $F \subseteq X^\omega$ mit Muller-Bedingung, A ist vollständig definiert, so akzeptiert $A, 2^Z \setminus \mathcal{F}$ die Menge $X^\omega \setminus F$, d.h. Abgeschlossenheit bzgl. Komplementbildung. Die Abgeschlossenheit bzgl. Vereinigung und Durchschnitt beweist man analog zur Theorie endlicher Wörter. □

Folgerung 7.7 *R_ω ist der Boolesche Abschluß von DR_ω, DR_ω ist Abgeschlossen bzgl. Vereinigung und Durchschnitt.*

Beweis. $F \subseteq X^\omega$ werde vom DEA $(A, \mathcal{F})$ mit Muller-Bedingung akzeptiert, . $F = \{Z_1, \ldots, Z_k\}$. F_i werde durch $(A, \{Z_i\})$ mit Muller-Bedingung akzeptiert, d.h. $F = \bigcup_{i=1}^k F_i$. Definiere die Menge $F(z)$ mit $z \in Z$ als Sprache die vom Automaten A mit $\{z\}$ mittels Büchi-Bedingung akzeptierte Menge. Es gilt $F_i = \bigcap_{z \in Z_i} F(z) \setminus \bigcup_{z' \notin Z_i} F(z)$, denn da A ist DEA, ist der Akzeptanzpfad eines Wortes eindeutig und da jedes $z \in Z_i$ mindestens einmal durchlaufen werden muß, folgt Durchschnitt über die Mengen $F(z)$,

und da F mittels Muller-Bedingung akzeptiert wird, folgt Differenz mit Vereinigung aller $F(z)$ mit $z \notin Z_i$, d.h.

$$F = \bigcup_{i=1}^{k} \left(\bigcap_{z \in Z_i} F(z) \setminus \bigcup_{z' \notin Z_i} F(z) \right)$$

Da $\bigcap_{z \in Z_i} F(z)$ und $\bigcup_{z' \notin Z_i} F(z)$ aus DR_ω sind, folgt die Behauptung. $\square$

7.1 Topologie in X^ω

Definition 7.2 (Metrik,Kugeln,offene Mengen,...) Eine Metrik ist eine Fkt. d, mit $d : M \times M \to \mathbb{R}$, die die folgenden Eigenschaften besitzt

(1) $d(x,y) \geq 0, d(x,y) = 0 \iff x = y$

(2) $d(x,y) = d(y,x)$

(3) $d(x,y) \leq d(x,z) + d(z,y)$

Die Fkt. $d(\xi,\eta) = \inf\{\frac{1}{q^n} | \xi(1)\ldots\xi(n) = \eta(1)\ldots\eta(n)\}$ definiert eine Metrik auf X^ω. Als Kugel definiere die Menge

$$K_\epsilon(\xi) = \{\eta | d(\xi,\eta) < \epsilon\} = \xi(1)\ldots\xi(n) \cdot X^\omega$$

mit $n = max\{l | q^{-l} > \epsilon\}$. Als offene Menge in (X^ω, d) definiere $F = W \cdot X^\omega$ mit $w \subseteq X^*$. Als G_δ-Mengen in (X^ω, d) bezeichne abzählbare Durchschnitte von offenen Mengen.

Bemerkung 7.11 (2) verhindert das Auftreten des Krokodilphänomens, d.h. daß das Krokodil vom Kopf zum Schwanz $12m$ lang ist, aber die Länge vom Schwanz zum Kopf nur $8m$ mißt (Rachen). (3) bedeutet, das man nicht abkürzen kann, wenn man den Umweg über z geht. In der obigen Definition besitzt inf die in der Mathematik übliche Bedeutung, d.h. inf M, wobei M eine Menge von Zahlen ist, steht in etwa für die kleinste Zahl, die in M enthalten ist, mathematisch nicht korrekt, aber inhaltlich richtig.

Satz 7.8 (Satz von Davis (1)) *Eine ω-Sprache $F \subseteq X^\omega$ ist genau dann eine G_δ-Menge, wenn es ein $W \subseteq X^*$ mit $F = \{\xi | |\{w | w \in W, w \sqsubseteq \xi\}| = \infty\}$ gibt, d.h. jedes Wort in W ist unendlich oft ein Präfix eines Wortes aus F.*

Beweis.

- F habe die angegebene Gestalt, zu konstruieren $(w_i)_{i \in \mathbb{N}}$ mit $F = \bigcap_{i \in \mathbb{N}} w_i X^\omega$. Setze $W_i = \{w | w \in W, w \text{ hat genau } i \text{ Präfixe in } W\}$.

- $F = \bigcap_{i=0}^{\infty} W_i X^\omega$, zu konstruieren $W \subseteq X^\omega$. Man kann immer erreichen, daß die W_i folgende Forderungen erfüllen:

  (a) $W_{i+1} X^\omega \subseteq W_i X^\omega$, andernfalls ersetze W_i durch $W_i^* = \bigcap_{j=1}^{i} W_j X^\omega$, es gilt, wie man leicht sieht, $F = \bigcap_{i=0}^{\infty} W_i^*$.

  (b) $w \in W_i \to |w| \geq i$, andernfalls ersetze W_i durch $W_i^* = \bigcup_{w \in W_i} w X^i$, es gilt, wie man leicht sieht, $W_i X^\omega = W_i^* X^\omega$.

(c) W_i ist präfixfrei, da für $w.v \in W_i$ mit $w \sqsubseteq v$ $wX^\omega \subseteq vX^\omega$ gilt, d.h. v kann entfernt werden, ohne W_iX^ω zu verändern.

Setzte nun $W = \bigcup_{i\in\mathbb{N}} W_i$, es gilt $F = \{\xi | |\{w|w \in W, w \sqsubseteq \xi\}| = \infty\}$, da $\forall \xi \in F$ und für jedes i ein $w_i \in W_i$ mit $w_i \sqsubseteq \xi$ existiert, und aufgrund Forderung (c) gibt es genau ein w_i. Da $|w_i| \geq i$ aber von endlicher Länge, hat ξ unendlich viele Präfixe in W, da $|w_i|$ von endlicher Länge kann es nur in endlich vielen W_i enthalten sein, d.h.ξ hat in unendlich vielen W_i ein Präfix, mit (a) folgt nun, ξ hat in jedem W_i ein Präfix.

□

Definition 7.3 Eine Tafel $\mathcal{F} \subseteq 2^Z$ einer Muller-Akzeptierungsbedingung heißt Filter, wenn aus $Z' \in \mathcal{F}$ und $Z' \subseteq Z''$ stets $Z'' \in \mathcal{F}$ folgt, d.h. $\mathcal{F}$ ist abgeschlossen nach oben bzgl. Inklusion.

Bemerkung 7.12 Es gibt Tafeln $\mathcal{F}$, die zu Filter erweiterbar sind ohne die Akzeptanzmenge zu verändern, dies ist nicht immer möglich. Aus $Z' \in \mathcal{F}$ und $Z' \not\emptyset$ folgt für einen Filter sofort $Z \in \mathcal{F}$.

Lemma 7.9 Eine ω-Sprache $F \subseteq X^\omega$ wird genau dann von einem DEA A Büchi akzeptiert, wenn sie von einem DEA B mit Filter $\mathcal{F}$ Muller akzeptiert wird.

Beweis. Sei $A = (X, Z, z_0, f)$, DEA mit Filter $\mathcal{F}$ der $F \subseteq X^\omega$ mit Muller-Bedingung akzeptiert. $F' = X^\omega \setminus F$ wird von A mit $2^Z \setminus \mathcal{F}$ mit Muller-Bedingung akzeptiert. Die Menge $2^Z \setminus \mathcal{F} = \{Z_1, \ldots Z_k\}$ ist bzgl. Inklusion abgeschlossen nach unten, in der Verbandstheorie nennt man eine solche Menge ein Ideal. Sei $B = (X, S, s_0, g), S' \subseteq S$ mit $\{S''|S'' \subseteq S'\}$ ein DEA der F' mit Muller-Bedingung akzeptiert, d.h. Automat fängt sich in S'. dann akzeptiert B mit $S \setminus S'$ die Menge F mit Büchi-Bedingung. Konstruktion des Automaten B zur Akzeptanz von F'. Da $\mathcal{F}$ Filter, gilt $\xi \in F' \iff$ für den Lauf η von ξ auf A gilt, es gibt ein j mit $\inf_A(\eta) \subseteq Z_j$. Setze $S = Z \times 2^{\{1,\ldots,k\}}$, $s_0 = \{z_0, \emptyset\}$ und

$$g\left((z, M), x\right) = \begin{cases} (f(z,x), M \cap \{i|f(z,x) \in Z_i\}) & : \quad M \neq \emptyset \\ (f(z,x), \{i|f(z,x) \in Z_i\}) & : \quad M = \emptyset \end{cases}$$

(Da $A, \mathcal{F}$ die Menge F mit der Muller-Bedingung akzeptiert, ist ab einem endlichen Zeitpunkt. die Indizemenge der 2. Komponente unverändert.) Setze nun weiterhin $S' = Z \times \left(2^{\{1,\ldots k\}} \setminus \{\emptyset\}\right)$. Zu zeigen $\xi \in F' \iff \inf_B(\eta(\xi)) \subseteq S'$, woraus für ein $\xi \in F$ die Existenz eines i folgen würde, mit $\inf_A(\eta) \subseteq Z_i$, wobei η Lauf von ξ auf A ist. Betrachtung des Laufes σ von ξ auf B, angenommen $\inf_B(\eta) \not\subseteq S'$, d.h. es gibt $t_k \quad k = 1, 2, \ldots$ mit $t_k < t_{k+1}$ und $\sigma(t_k) = (\eta(t_k), \emptyset)$. Nach Konstruktion gilt $\sigma(t'+1) = (\eta(t'+1), M_{t'+1})$ mit $i \in M_{t'+1}$, weiter gilt auch $\sigma(t'+l) = (\eta(t'+l), M_{t'+l})$ mit $i \in M_{t'+l}$ für alle $l > 0$, d.h. es kann nicht unendlich viele t_k mit $\sigma(t_k) = (\eta(t_k), \emptyset)$ geben, d.h. Annahme ist falsch, d.h. $\inf_B(\sigma) \subseteq S'$, d.h. es gibt einen Zeitpunkt t mit $\sigma(j) = (\eta(j), M_j), M_j \neq \emptyset \quad \forall j \geq t$, mithin $\inf_A(\eta) \subseteq Z_i$ für ein i. □

Satz 7.10 (Satz von Davis (2))

$$DR_\omega = R_\omega \cap G_\delta$$

Beweis. ‚$\subseteq$' kann mittels Übungsaufgabe 13.1 und des 1.Satzes von Davis bewiesen werden, m.a.W. es wird hier nur die Richtung ‚$\supset$' bewiesen. F werde vom DEA $A = (x, Z, z_0, f), \mathcal{F}$ mit der Mullerbedingung akzeptiert, A sei vollständig definiert. Da F eine G_δ-Menge gibt es $W \subseteq X^*$ mit $F = \{\xi | \xi \in X^\omega, |\{w | w \in W, w \sqsubset \xi\}| = \infty\}$. Zu zeigen ist, das $\mathcal{F}$ zu einem Filter erweiterbar ist, ohne die Akzeptanzmenge zu verändern. Konzentration auf stark zusammenhängende Komponenten von A, d.h. (Stark zusammenhängend : Teilmenge S eines gerichteten Graphen G , in dem jeder Knoten aus S durch einen vollständig in S gelegengen Pfad erreichbar ist.) Es gilt: Aus $Z' \subseteq Z$ ist nicht stark zusammenhängend folgt, die akzeptierte ω-Sprache von A und $\{Z'\}$ ist leer, bei Verwendung der Muller-Bedingung. Ersetze $\mathcal{F}$ durch $\mathcal{F}_1 = \mathcal{F} \cap \mathcal{T}$ mit

$$\mathcal{T} = \{Z' | Z' \subseteq Z, Z' \text{ von } z_0 \text{ erreichbare stark zusammenhängende Komponente von } A\}$$

Hierbei wurde die obige Bemerkung nur formalisiert. Setze nun

$$\mathcal{F}_2 = \mathcal{F}_1 \cup \{Z'' | Z'' \in \mathcal{T}, \exists Z' \in \mathcal{F}_1 \text{mit} Z' \subseteq Z''\}$$

d.h. fülle $\mathcal{F}_1$ zum Filter auf. Die Akzeptanzmenge wird nicht verändert, durch die Ersetzung von $\mathcal{F}_1$ durch $\mathcal{F}_2$. (Warum ??) Zu zeigen $\mathcal{F}_2$ ist Filter, angenommen dies ist falsch, d.h. $\exists Z_1, Z_2 \in \mathcal{T}, Z_1 \in \mathcal{F}_2, Z_2 \notin \mathcal{F}_2$. Nach Definition von $\mathcal{T}$ gibt es $e \neq u \neq v$ mit $z = f(z_0, w) = f(z, v) = f(z, u)$, und sei nun o.B.d.A. $Z_1 = \{f(z, v') | v' \sqsubseteq v\}, Z_2 = \{f(z, u') | u' \sqsubseteq u\}$. $z_1 \in \mathcal{F}_2 \rightarrow wu, v^* v^\omega \in F$, da $Z_2 \notin \mathcal{F}_2$ folgt $w(v^* u)^\omega \cap F = \emptyset$. Aus $wv^\omega \in F$ folgt Existenz von $w_1 \in W$ und $n_1 \in \mathbb{N}$ mit $w_1 \sqsubseteq wv^\omega, wv^{n_1} uv^\omega \in F$, somit gibt es $w_2 \in W, n_2 \in \mathbb{N}$ mit $wv^{n_1} u \sqsubseteq w_2 \sqsubseteq wv^{n_1} uv^{n_2}, wv^{n_1} uv^{n_2} uv^\omega \in F$, somit gibt es .... Man erhält $\xi = wv^{n_1} uv^{n_2} uv^{n_3} \ldots, w_1 \sqsubseteq w_2 \sqsubseteq w_3 \cdots \sqsubseteq \xi, w_i \in W \rightarrow \xi \in F$, aber $\inf(\eta) = Z_2$ mit η Lauf von A auf ξ (da A DEA ist Lauf eindeutig), d.h. $\xi \notin F$, Widerspruch. $\square$

A Übungsaufgaben

Aufgabe 1.1:

Es sei $(H, \circ)$ eine Halbgruppe. Zeigen Sie:

a) Gibt es in H eine Linkseins und eine Rechtseins, so existiert in H auch ein Einselement und dieses ist eindeutig bestimmt.

b) Gibt es in H eine Linksnull und eine Rechtsnull, so existiert in H auch ein Nullelement und dieses ist eindeutig bestimmt.

Aufgabe 1.2:

Es sei A eine Menge und 2^A die Potenzmenge von A. Zeigen Sie, daß 2^A mit der Durchschnittsoperation eine Halbgruppe bildet. Bestimmen Sie Eins-, Null-, und reguläre Elemente.

Aufgabe 1.3:

a) Geben Sie eine reguläre Halbgruppe an, die keine Gruppe ist.

b) Konstruieren Sie eine Halbgruppe mit 5 Elementen, die alle Rechtseinsen und Linksnullen sind.

Aufgabe 1.4:

Es sei $(H, \circ)$ eine Halbgruppe und R die Menge aller rechtsregulären Elemente von H. Zeigen Sie, daß $(R, \circ)$ eine Halbgruppe ist, sofern R nicht leer ist.

Definitionen

Es seien H eine nichtleere Menge und $\circ : H \times H \to H$ eine Abbildung. Das Paar $(H, \circ)$ heißt *Halbgruppe*, falls $\circ$ assoziativ ist, d.h. wenn

$$\forall x \forall y \forall z (x, y, z \in H \to x \circ (y \circ z) = (x \circ y) \circ z) \text{ gilt.}$$

Im folgenden sei $(H, \circ)$ eine Halbgruppe.

$$\begin{aligned}
y \in H \text{ heißt } \textit{Rechtseins} &:\Longleftrightarrow \quad \forall x (x \in H \to x \circ y = x) \\
x \in H \text{ heißt } \textit{Linkseins} &:\Longleftrightarrow \quad \forall y (y \in H \to x \circ y = y) \\
y \in H \text{ heißt } \textit{Einselement} &:\Longleftrightarrow \quad y \text{ ist sowohl Rechts- als auch Linkseins} \\
y \in H \text{ heißt } \textit{Rechtsnull} &:\Longleftrightarrow \quad \forall x (x \in H \to x \circ y = y) \\
x \in H \text{ heißt } \textit{Linkssnull} &:\Longleftrightarrow \quad \forall y (y \in H \to x \circ y = x) \\
y \in H \text{ heißt } \textit{Nullelement} &:\Longleftrightarrow \quad y \text{ ist sowohl Rechts- als auch Linksnull} \\
y \in H \text{ heißt } \textit{rechtsregulär} &:\Longleftrightarrow \quad \forall x \forall z (x, z \in H \wedge x \circ y = z \circ y \to x = z) \\
x \in H \text{ heißt } \textit{linksregulär} &:\Longleftrightarrow \quad \forall y \forall z (y, z \in H \wedge x \circ y = x \circ z \to y = z) \\
y \in H \text{ heißt } \textit{regulär} &:\Longleftrightarrow \quad y \text{ ist sowohl rechts- als auch linksregulär} \\
x \in H \text{ heißt } \textit{idempotent} &:\Longleftrightarrow \quad x \circ x = x
\end{aligned}$$

Sind alle Elemente von H regulär, so heißt $(H, \circ)$ *regulär*.

Aufgabe 2.1:

Es seien M eine Menge und M^M die Menge aller Abbildungen von M in sich selbst. Das Paar $(M^M, \circ)$, wobei $\circ$ die Hintereinanderausführung von Abbildungen ist, bildet eine Halbgruppe. Formal definieren wir $\circ$ als

$$(f \circ g)(x) = g(f(x)) \text{ für alle } x \in M.$$

1. Es sei A eine *endliche* Menge. Bestimmen Sie in der Halbgruppe $(A^A, \circ)$ Rechtseinsen, Rechtsnullen sowie rechtsreguläre und idempotente Elemente.
2. Bestimmen Sie in der Halbgruppe $(\mathbb{N}^{\mathbb{N}}, \circ)$ Rechtseinsen, Rechtsnullen sowie rechtsreguläre und idempotente Elemente.

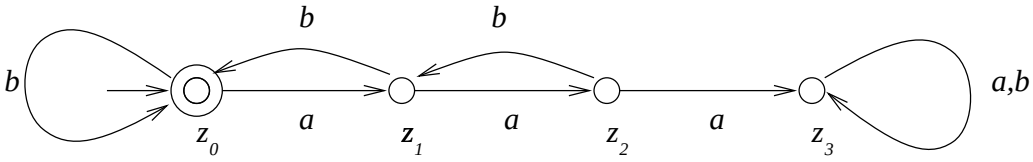
Aufgabe 2.2:

Ein Wort $v \in X^*$ heißt *berandet*, falls $v = e$ gilt oder es ein Präfix $w \sqsubseteq v$, $e \neq w \neq v$, von v gibt, welches zugleich Suffix von v ist. Anderenfalls heißt v *unberandet* (englisch: unbordered). Zeigen Sie:

1. $v \in X^* \setminus \{e\}$ ist genau dann berandet, wenn es $u \in X^* \setminus \{e\}, w \in X^*$ mit $v = uwu$ gibt.
2. Ist v unberandet und $v \sqsubseteq u \cdot v$ für $u \neq e$, so folgt $v \sqsubseteq u$.
3. Es sei $a \in X$ nicht der erste Buchstabe von $u \neq e$. Dann ist $v := u \cdot a^{|u|}$ unberandet. Ist u unberandet und $a \in X$ nicht der erste Buchstabe von u, so ist auch ua unberandet.
4. Jedes unberandete Wort ist primitiv. Gilt auch die Umkehrung?

Aufgabe 3.1:

Bestimmen Sie die Elemente des Transitionsmonoids für den folgenden DEA. Ermitteln Sie die idempotenten Elemente und die Rechtsnullen.



Aufgabe 3.2:

Es seien $v \in X^*$ ein unberandetes Wort, $V := X^* \setminus X^* \cdot v \cdot X^*$ die Menge aller Wörter, die v nicht als Infix enthalten. Man zeige, daß dann $C := V \cdot v$ ein Präfix-Code ist.

Aufgabe 3.3:

Stellen Sie fest, ob die folgenden Mengen Codes sind:

1. $\{01, 011, 10, 1000, 1100, 0111\}$
2. $\{1010, 001, 101, 0001, 1101, 1011\}$

Aufgabe 3.4:

Konstruieren Sie einen Präfix-Code über einem dreielementigen Alphabet, der ein Wort der Länge 1, vier Wörter der Länge 2 und fünf Wörter der Länge 3 enthält.

Aufgabe 4.1:

Konstruieren Sie einen binären Code für die Verteilung

$$\mathcal{P} = \{0.3, 0.18, 0.15, 0.15, 0.07, 0.06, 0.05, 0.04\}.$$

Aufgabe 4.2:

Gegeben sei eine Quelle mit den Symbolen a_1, a_2 und a_3, deren relative Häufigkeiten 0.6, 0.3 und 0.1 betragen. Konstruieren Sie einen längenvariablen Präfix-Code und berechnen Sie die mittlere Codewortlänge.
Betrachten Sie nun dieselbe Quelle und codieren Sie nach demselben Verfahren wie zuvor Paare von Quellsymbolen $a_i a_j$, $1 \leq i, j \leq 3$, wobei deren relative Häufigkeit $p_i p_j$ betragen soll. Berechnen Sie auch hier die mittlere Codewortlänge. Welches Verfahren ist günstiger?

Aufgabe 4.3:

Stellen Sie fest, ob es unendliche ternäre Codes (d.h. über einem dreielementigen Alphabet) mit

1. je 2^n Wörtern der Länge n für $n \geq 2$,
2. je 2^{n-1} Wörtern der Länge n für $n \geq 1$ gibt.

Aufgabe 4.4:

1. Konstruieren Sie einen ternären Code mit der Längenverteilung $\{1, 1, 2, 3, 3, 3\}$.
2. Konstruieren Sie nach dem Verfahren von Shannon einen ternären Code zur Wahrscheinlichkeitsverteilung $\{1/3, 1/3, 1/9, 1/27, 1/27, 1/27, 1/27\}$.

Aufgabe 5.1:

Konstruieren Sie ternäre Huffman-Codes für die folgenden Verteilungen. Bestimmen Sie jeweils die mittlere Codewortlänge und die Entropie.

1. $\mathcal{P} = \{0.4, 0.3, 0.1, 0.1, 0.04, 0.03, 0.02, 0.01\}$
2. $\mathcal{P} = \{0.15, 0.14, 0.13, 0.11, 0.11, 0.10, 0.10, 0.09, 0.07\}$

Aufgabe 5.2:

Gegeben sei eine Quelle mit den Symbolen a_1, a_2 und a_3, deren relative Häufigkeiten $p_1 = 0.5$, $p_2 = 0.3$ und $p_3 = 0.2$ betragen.
Betrachten Sie nun dieselbe Quelle und codieren Sie nach dem Verfahren von Huffman Paare von Quellsymbolen $a_i a_j$, $1 \leq i, j \leq 3$, wobei für deren relative Häufigkeit p_{ij}

1. $p_{ij} = p_i p_j, 1 \leq i, j \leq 3$,
2. $p_{11} = 0.10$, $p_{12} = 0.30$, $p_{13} = 0.00$, $p_{21} = 0.20$, $p_{22} = 0.01$, $p_{23} = 0.07$, $p_{31} = 0.30$, $p_{32} = 0.01$, $p_{33} = 0.01$

gelte. Berechnen Sie jeweils die mittlere Codewortlänge und die Entropie.

Aufgabe 5.3:

Konstruieren Sie nach dem Algorithmus aus der Vorlesung einen binären alphabetischen Code zur folgenden Verteilung. Bestimmen Sie die mittlere Codewortlänge und die Entropie.

$$\mathcal{P} = \{0.05, 0.1, 0.25, 0.1, 0.06, 0.03, 0.01, 0.2, 0.16, 0.04\}$$

Aufgabe 6.1:

Es sei $n \geq 1$ eine natürliche Zahl. Weiter seien I und J Mengen natürlicher Zahlen, so daß für $i, i' \in I$ und $j, j' \in J$ aus

$$i + j \equiv i' + j' \ (\mod n)$$

stets $i = i' \wedge j = j'$ folgt. Es sei $V = \{a^i b a^j : i \in I, j \in J\}$. Zeigen Sie, daß $C = V \cup \{a^n\}$ ein Code ist.

Aufgabe 6.2:

Zeigen Sie, daß eine Menge $C = \{w, v\} \subseteq X^* \setminus \{e\}$ genau dann ein Code ist, wenn $w^{|v|} \neq v^{|w|}$ gilt.

Aufgabe 6.3:

Entscheiden Sie, ob die folgenden endlichen Sprachen Codes sind:

1. $\{100, 1001, 11010, 11011, 0101, 101, 011011\}$
2. $\{111, 10011, 1001, 00111, 001, 110, 11010\}$

Aufgabe 6.4:

Zeigen Sie, daß ein endlicher Präfix-Code $C = \{w_1, \ldots, w_n\} \subseteq X^* \setminus \{e\}$ genau dann maximal ist, wenn die Beziehung

$$\sum_{i=1}^{n} (AnzX)^{-|w_i|} = 1$$

erfüllt ist.

Aufgabe 7.1:

Es seien $n > 0$ und G eine durch

$$S \to a \mid b \cdot S^n$$

gegebene kontextfreie Grammatik. Zeigen Sie, daß die Sprache $L(G)$ ein Präfix-Code ist.

Aufgabe 7.2:

Jeder endliche Präfixcode $C \subseteq X^*$ ist in einem maximalen endlichen Präfixcode $\hat{C} \subseteq X^*$ enthalten.

Aufgabe 7.3:

Zeigen Sie, daß folgendes Problem entscheidbar ist: Gegeben sei eine reguläre Grammatik G; ist $L(G)$ ein vollständiger Code?

Aufgabe 7.4:

Ein Code $C \subseteq X^*$ heißt *voll*, wenn es zu jedem $w \in X^*$ eine eindeutig bestimmte Folge $v_1, v_2 \ldots, v_n$ von Codewörtern und ein Präfix $u \in X^*$ eines Codewortes mit $w = v_1 v_2 \cdots v_n u$ gibt. Zeigen Sie:

1. Jeder volle Code ist ein Präfix-Code.
2. Ein endlicher Präfix-Code mit der maximalen Codewortlänge m ist genau dann voll, wenn jedes Wort $w \in X^m$ ein Codewort ist oder ein Codewort als Präfix besitzt.

Aufgabe 8.1:

Zeigen Sie, daß ein magerer Code, der maximal in der Klasse der Codes mit beschränkter Decodierverzögerung ist, auch maximal in der Klasse aller Codes ist.

Aufgabe 8.2:

Hat ein Code C eine Decodierverzögerung von höchstens m für jedes $w \in C$, so ist keine C-Kette länger als $2m + 2$.

Aufgabe 8.3:

Ist für einen Code C jede C-Kette nicht länger als ℓ, so hat C eine Decodierverzögerung, die durch $\ell - 1$ beschränkt ist.

Aufgabe 8.4:

Es sei p eine Primzahl und $i \in \{1, \ldots, p-1\}$. Zeigen Sie

1. Es existiert genau ein $j \in \{1, \ldots, p-1\}$ mit $i \cdot j \equiv 1 \pmod{p}$.
2. Es gilt $i^2 \equiv 1 \pmod{p}$ genau dann, wenn $i = 1$ oder $i = p - 1$.

Wir fixieren die folgende spezielle rekursive Menge von Paaren, bei denen die Zusatzinformation gewissermaßen die Wortlänge ist:[1]

$$\mathcal{X}_\ell := \{(w, \mathcal{Q}^{-1}(|w|)) : w \in X^*\} \subseteq X^* \times X^* \ .$$

Aufgabe 9.1:

Es sei $M \subseteq \mathcal{X}_\ell$ eine Menge, für welche $\mathcal{X}_\ell \setminus M$ rekursiv aufzählbar ist, weiter sei $M_v := \{w : (w, v) \in M\}$. Zeigen Sie[2] (unter Beachtung von $|\log_{\mathrm{card}\, X} 0| = \infty$.)

$$\exists c \forall w (K(w/\mathcal{Q}^{-1}(|w|)) < |\log_q \mathrm{card} M_{\mathcal{Q}^{-1}(|w|)}| + c) \ .$$

Aufgabe 9.2 (Folgerung 6.9.1 der Vorlesung)**:**

Zeigen Sie, daß für eine beliebige Familie $(w_i)_{i\in\mathbb{N}}$, wobei $|w_i| \geq i$ gilt, die Komplexität $K(w_i/v)$ bei festem v unbeschränkt wächst.

Aufgabe 9.3 (Folgerung 6.13.1 der Vorlesung)**:**

Weisen Sie nach, daß die Funktion $\kappa\colon \mathbb{N} \to \mathbb{N}$, wobei $\kappa(n) := \min\{K(\mathcal{Q}^{-1}(m)) : m \geq n\}$, monoton nicht fallend und unbeschränkt ist.

Aufgabe 9.4:

Eine Präfixcodierung ist eine Abbildung $\varphi\colon X^* \to X^*$, für die kein Bild (Codewort) $\varphi(w)$ Anfangsstück eines anderen Codewortes $\varphi(v), v \neq w$ ist. Man konstruiere eine Präfixcodierung mit der Längenbeschränkung $|\varphi(w)| \leq |w| + 2 \cdot \log_q |w| + const.$.
Wir betrachten ein Alphabet X mit q $(\infty > q \geq 2)$ Buchstaben.

Aufgabe 10.1:

Man zeige: $\exists c \forall w \forall w' (w, w' \in X^* \to K(w \cdot w') \leq K(w') + |w| + 2 \cdot \log_q |w| + c)$.

Aufgabe 10.2:

Es sei $\varphi : X^* \to X^*$ eine injektive partiell-rekursive Funktion. Zeigen Sie, daß für alle $w \in \mathrm{dom}(\varphi)$ und für alle $v \in X^*$ die Beziehung $|K(w/v) - K(\varphi(w)/v)| \leq const$ gilt.

Aufgabe 10.3:

Weisen Sie die Gültigkeit der folgenden beiden Ungleichungen nach:

$$\begin{aligned} \exists c \forall w \forall v (K(w) &\leq 2 \cdot K(v) + K(w/v) + c) \\ \exists c \forall w \forall v (K(w) &\leq K(v) + 2 \cdot K(w/v) + c) \end{aligned}$$

Wir nennen eine Sprache $L \subseteq X^*$ *mager*, falls es eine Zahl $r < q$ derart gibt, daß für fast alle natürlichen Zahlen n die Anzahl der Wörter der Länge n in L nicht größer als r^n ist. Wir

[1] $\mathcal{Q}$: $X^* \to \mathbb{N}$ sei dabei eine *quasilexikographische* Ordnung von X^*.

[2] Hinweis: Es wird Ihnen nichts weiter übrig bleiben, als das relative Komplement $\mathcal{X}_\ell \setminus M$ aufzuzählen; die Frage ist nur, wie lange.

nennen

$$H_L := \inf\{r : r \geq 0 \wedge \lim_{n\to\infty} \frac{\text{card } L \cap X^n}{r^n} = 0\}$$

die *Entropie* der Sprache L.

Aufgabe 10.4:

Es sei $L \subseteq X^*$ eine magere rekursive Sprache. Dann gilt

$$\forall\varepsilon(\varepsilon > 0 \to \exists c\forall w(w \in L \to K(w) \leq (H_L + \varepsilon) \cdot |w| + c)) \ .$$

Wir betrachten ein Alphabet X mit q $(\infty > q \geq 2)$ Buchstaben.
Wir nennen eine Sprache $L \subseteq X^*$ *mager*, falls es eine Zahl $r < q$ derart gibt, daß für fast alle natürlichen Zahlen n die Anzahl der Wörter der Länge n in L nicht größer als r^n ist. Wir nennen

$$H_L := \inf\{\log_q r : r \geq 1 \wedge \lim_{n\to\infty} \frac{\text{card } L \cap X^n}{r^n} = 0\}$$

die *Entropie* der Sprache L.

Aufgabe 11.1:

Es sei $L \subseteq X^*$ eine magere rekursive Sprache. Dann gilt

$$\forall\varepsilon(\varepsilon > 0 \to \exists c\forall w(w \in L \to K(w) \leq (H_L + \varepsilon) \cdot |w| + c)) \ .$$

Aufgabe 11.2:

Zeigen Sie, daß für alle $n \in \mathbb{N}$ ein n' aus dem Intervall $[n - \sqrt{n}, n + \sqrt{n}]$ existiert, so daß

$$K'(n') \geq \frac{n}{2} - c$$

gilt, wobei c eine geeignete, von n unabhängige Konstante ist.

Aufgabe 11.3:

Zeigen Sie, daß für alle $n \in \mathbb{N}$ ein n'' aus dem Intervall $[n - \sqrt{n}, n + \sqrt{n}]$ existiert, so daß

$$K'(n'') \leq \frac{n}{2} + c$$

gilt, wobei c eine geeignete, von n unabhängige Konstante ist.

Aufgabe 11.4:

Für $\zeta \in X^\omega$ und $n \in \mathbb{N}$ definieren wir $K_0(\zeta/n) := K(w)$, wobei w das Präfix der Länge n von ζ ist.
Zeigen Sie: Ist ζ rekursiv, so gilt $K_0(\zeta/n) \leq \log_q n + c$ für alle $n \in \mathbb{N}$.

Aufgabe 12.1:

Geben Sie jeweils einen Büchi-, einen Rabin- und einen Muller-Automaten an, der die ω-Sprache $E = \{a, b\}^* a^\omega$ akzeptiert.

Aufgabe 12.2:

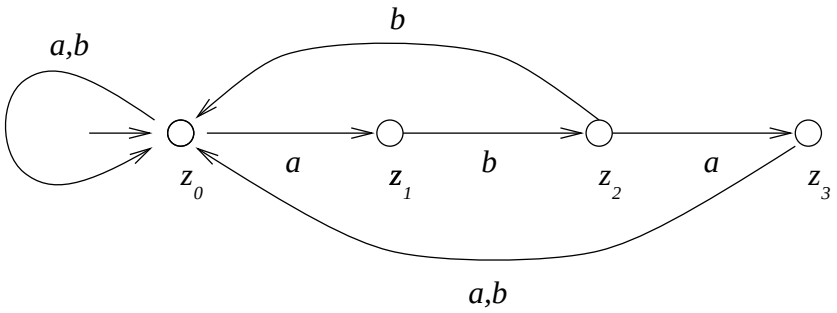
Geben Sie jeweils einen Büchi-, einen Rabin- und einen Muller-Automaten an, der genau die ω-Wörter $\xi \in \{a, b, c\}^\omega$ akzeptiert, für die folgende Bedingung erfüllt ist: Tritt a unendlich oft in ξ auf, so folgt in ξ auf b stets c.

Aufgabe 12.3:

Geben Sie jeweils einen Büchi-, einen Rabin- und einen Muller-Automaten an, der die ω-Sprache $E = (b^* a)^\omega \cup (a^2)^* c a^\omega$ akzeptiert.

Aufgabe 12.4:

Bestimmen Sie die von dem folgenden Rabin-Automaten akzeptierte Sprache; dabei sei $\mathcal{T} = \{(\{z_3\}, \{z_2\})\}$.



Aufgabe 13.1:

Eine ω-Sprache $F \subseteq X^\omega$ wird von einem DEA genau dann Büchi-akzeptiert, wenn es eine reguläre Sprache $W \subseteq X^*$ mit der Eigenschaft

$$\xi \in F \iff \text{card}\{w : w \sqsubset \xi \wedge w \in W\} = \infty$$

gibt.

Aufgabe 13.2:

Es sei W eine reguläre Sprache. Zeigen Sie, daß $\{\xi : \forall w(w \sqsubset \xi \to w \in W)\}$ durch einen DEA Büchi-akzeptiert wird.

Aufgabe 13.3:

Geben Sie die Nerode-Rechtskongruenz für folgende ω-Sprachen an:

1. $\{a, b\}^* a^\omega \cup c a^\omega$,
2. abc^ω,
3. V^ω, wobei V durch die Grammatik $S \to a \mid bSS$ erzeugt wird.

Aufgabe 13.4:

Welche der in Aufgabe 13.3 gegebenen ω-Sprachen wird von dem von ihrer Nerode-Kongruenz $\simeq$ abgeleiteten Automaten $\mathcal{A}_{\simeq}$ akzeptiert?

B Lösungen ausgewählter Aufgaben

Aufgabe 2.1
Sei M eine Menge, $|M| < \infty$ oder $M = \mathbb{N}$. Betrachte die Halbgruppe $(M^M, \circ)$. Dabei sei $f \circ g(x) = f(g(x))$.

(i) Linkseinsen, $E_L = \{f \in M^M : f \circ g = g \quad \forall g \in M^M\}$:
Offensichtlich ist $\mathrm{id} \in E_L$. Sei $f \in E_L$. Setze $g = \mathrm{id}$. Es ist

$$f(x) \underset{g=id}{=} f(g(x)) \underset{f \in E_L}{=} g(x) = x.$$

Also ist $E_L = \{\mathrm{id}\}$, d.h. die einzige Linkseins ist die Identität auf M.

(i)' Rechtseinsen, $E_R = \{f \in M^M : g \circ f = g \quad \forall g \in M^M\}$:
Sei $f \in E_R$. Setze $g = \mathrm{id}$. Dann ist $g \circ f = f = \mathrm{id}$. Also ist $E_R = \{\mathrm{id}\}$.

(ii) Linksnullen $N_L = \{f \in M^M : f \circ g = f \quad \forall g \in M^M\}$:
Sei $f \in N_L, x_1 \in M$ beliebig. Setze $g(x) \equiv x_1$. Dann ist

$$f(x) \underset{f \in N_L}{=} f(g(x)) = f(x_1),$$

d.h. f ist konstante Funktion, $N_L = \{f \in M^M : f \equiv \mathrm{const}\}$.

(ii)' Rechtsnullen, $N_R = \{f \in M^M : g \circ f = f \quad \forall g \in M^M\}$:
Sei $f \in M^M$ und $x \in M$ beliebig. Wähle g mit $g(f(x))$ nicht fix. Dann ist $f \notin N_R$, d.h. $N_R = \emptyset$.

(iii) linkssreguläre Elemente $R_L = \{f \in M^M : f \circ g = f \circ h \Rightarrow g = h\}$:
Setze $R_L{}' = \{f : f \text{ ist injektiv}\}$. Zeige $R_L = R_L{}'$. Sei zunächst $f \in R_L{}'$. Aus $f(g(x)) = f(h(x))$ folgt $g(x) = h(x)$ für alle x, also $g = h$, somit ist $f \in R_L$.
Sei jetzt $f \in R_L{}', x, y \in M$. Setze $g(z) \equiv x, h(z) \equiv y$. Aus $f(x) = f(y)$ folgt

$$f(g(z)) = f(h(z)) \underset{f \in R_L}{\Rightarrow} g(z) = h(z) \Rightarrow x = y,$$

d.h. f ist injektiv.

(iii)' rechtsreguläre Elemente $R_R = \{f \in M^M : g \circ f = h \circ f \Rightarrow g = h\}$:
Zeige $R_R = \{f : f \text{ ist surjektiv}\}$. Sei f surjektiv. Aus $g(f(x)) = h(f(x)) \forall x \in M$ folgt dann $g(z) = h(z) \forall z \in M$, d.h. $f \in R_R$.
Sei $f \in R_R$. Angenommen f nicht surjektiv. Insbesondere ist dann $|M| > 1$. Dann gibt es Funktionen $g \neq h$, die auf $f(M)$ übereinstimmen. Also ist $f \notin R_R$, Widerspruch.

(iv) idempotente Elemente $I = \{f \in M^M : f \circ f = f\}$:
Setze $I' = \{f : \exists N \subseteq M : f_{|N} = \mathrm{id}, f(M) = N\}$. Zeige $I = I'$. Sei $f \in I$. Setze $N = f(M)$. Es ist $f_{|N} = \mathrm{id}$, also $f \in I'$. Sei nun $f \in I'$. Dann ist $f(f(x)) = f_{|M}(f(x)) = f(x)$, also $f \in I$.

Aufgabe 2.2

(a) Sei $v = uwu$. Nach Definition ist v berandet. Sei nun v berandet, d.h. $\exists r_1, r_2, w \in X^* : v = wr_1 = r_2w$. Mit Lemma 1.8 folgt $\exists s, t \in X^* : w = (st)^k s,\ r_1 = ts,\ r_2 = st$, also ist $v = st(st)^k s$ und es gilt (a).

(b) Angenommen $|v| > |u|$. Wegen $v \sqsubseteq uv$ gibt es ein Präfix $v_1 \neq e$ von v mit $v = uv_1$. Also ist v_1 auch Suffix von v. Mit (a) ist v berandet, ein Widerspruch.

(c) Sei $a \not\sqsubseteq u$ und $v = ua^{|u|}$. Falls v berandet, so ist mit (a) $v = xwx, x = a^n$, also $a \sqsubseteq u$, ein Widerspruch. Sei v unberandet und $a \not\sqsubseteq v$. Falls va berandet, so gilt mit (a) $va = xwx$, mit $x = x_1a$, $x_1 \neq e$. Folglich ist $v = x_1awx_1$, also ist v berandet, Widerspruch.

(d) Sei v nicht primitiv, d.h. $v = u^n$, $n > 1$. Insbesondere ist dann $v = uxu$, also ist v berandet. Die Umkehrung gilt nicht, z.B. ist „Solaris“ berandet und primitiv.

Literatur

[1] J. Adámek, *Foundations of Coding*, Wiley & Sons, Chichester, 1991.

[2] J. Berstel and D. Perrin, *Theory of Codes*, Academic Press, Orlando, 1985.

[3] C. Calude, *Information and Randomness. An Algorithmic Perspective*, Springer–Verlag, Berlin, 1994.

[4] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest, *Introduction to algorithms*, MIT-Press, Cambridge, Mass., 1995.

[5] S. Eilenberg, *Automata, Languages and Machines*, vol. A, Academic Press, New York, 1974.

[6] J. Engelfriet and H.J. Hoogeboom, *X-automata on ω-words*, Theoret. Comput. Sci **110** (1993), 1–51.

[7] W. Heise and P. Quattrocchi, *Informations- und Codierungstheorie*, Springer–Verlag, Berlin, 1989.

[8] M. Li and P.M.B. Vitányi, *An Introduction to Kolmogorov Complexity and its Applications*, Springer–Verlag, New York, 1993.

[9] W. Thomas, *Automata on infinite objects*, Handbook of Theoretical Computer Science (J. van Leeuwen, ed.), vol. B, Elsevier, Amsterdam, 1990, pp. 133–191.

[10] B.A. Trakhtenbrot and Ya.M. Barzdin, *Finite Automata, Behavior and Synthesis*, Nauka, Moscow, 1970, English tanslation: North Holland, Amsterdam, 1973.

Index

B^++, 2
B^{*1}, 2
DR_ω, 47
G_δ-Mengen, 48
R_ω, 47
X^ω, 43
inf(η), 43
κ-Fkt., 38
$\sqrt{M}$, 5
$\sqsubseteq$, 6

Aquivalenzrelation, 3

Beispiel von Markov, 17, **24**
Bernoulli Mas, 11
Buchi Akzeptierung, 43

C-Kette, 17
Code, 10
Code
    decodierbarer, 10
    Maximaler, 17
Codierung
    alphabetische, 15
    DES, 27
    endliche Substitution, 28
    One Time Pad, 30
    RSA, 34
    Skytale von Sparta, 26
    von Caesar, 26
    von Huffman, 14
    von Shannon, 13
    von Vigenere, 28

Decodierverzogerung, 24

endlicher Automat, 8
Entropie, 12
Erzeugendensystem , 2
    freies, 3
Eulersche φ-Funktion, 31, **31**

Filter, 49

Halbgruppe, 51
Halbgruppe , 2
    freie, 3
Homomorphismus, 2

idempotent, 51
Information, 12

Koinzidenzindex, 29
Kolmogorov Komplexitat, **35**, 36, 38
Kongruenzrelation, 3
Konigsches Unendlichkeits Lemma, 42
Krokodilphanomen, 48
Kugel, 48

Lange, 4
Lauf , 43
    vollstandiger, 43
Linkseins, 51
Linksnull, 51
linksregular, 51

Metrik, 48
Modulare Potenzierung, 33
Monoid, 2
Muller Akzeptierung, 43

Nerode Relation, 9
Nerodsche Rechtskongruenz, 45
Nult, 46

offene Menge, 48
Ordnungsrelation
    lexikographische, 10
    quasilexikographische, 10

prafix Abgeschlossenheit, 12
Prafix-Code, 10
    maximaler, 54
primitive Wurzel, 7
Primzahltest
    Fermat-Test, 33

Q-min, 37

Rabin Akzeptierung, 43
Rechtseins, 51
Rechtsnull, 51
rechtsregular, 51

regular, 51
regulare ω-Sprachen, 45

SAFRA Baum, 47
SAFRA-Konstruktion, 46
Satz
    Chinesischer Restsatz, 31
    uber Quellenkodierung, 15
    von Buchi, 44
    von C.F.Gaus, 34
    von Davis (1), 48
    von Davis (2), 49
    von Ehrenfeucht und Rozenberg, 21
    von Fermat, 31
    von Fermat und Euler, 32
    von Kleene, Myhill, Nerode, 9
    von Lévi, 5
    von Mac Millan, 11
    von McNaughton, 47
    von minimaler mittlerer Codewortlange, 13
    von P. Martin-Lof (1), 40
    von P. Martin-Lof (2), 40
    von P. Martin-Lof (3), 41
    von Sardinas und Peterson, 18
    von Shannon, 13
Sprache
    dichte, 20
    magere, 20
    vollstandige, 20
Suffix-Code, 10
syntaktische Relation, 9

T(L), 20
Transitionsmonoid von $\mathbb{A}$, 8

Ungleichung von Kraft, 10
Unterhalbgruppe, 2
Untermonoid, 2
unzerlegbar, 4

Wiedervereinigungslemma, 47
Wort
    berandetes, unberandetes, 52
    primitives, 7
    zirkulares, 7