INFORMATIQUE THÉORIQUE ET APPLICATIONS

H. JÜRGENSEN
M. KUNZE

Über die implementierung redundanzfreier Codes zur datenverschlüsselung

Informatique théorique et applications , tome 20, n° 1 (1986), p. 5-29
<http://www.numdam.org/item?id=ITA_1986__20_1_5_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 20, n° 1, 1986, p. 5 à 30)

ÜBER DIE IMPLEMENTIERUNG REDUNDANZFREIER CODES ZUR DATENVERSCHLÜSSELUNG (*) (**)

by H. JÜRGENSEN ([1]) and M. KUNZE ([2])

Communicated by J. BERSTEL

Abstract. — *The implementation of redundancy-free encodings is studied. Necessary and sufficient conditions are given for the complexity of encoding the length n output of a finite Markov source to be $O(n^3/^2)$ with arbitrarily large probability rather than the worst case of $O(n^2)$.*

Résumé. — *On étudie l'implementation des codes sans redondance et la complexité de coder les messages de longueur n d'une source de Markov finie. On démontre que cette complexité est $O(n^2)$ et on établit des conditions nécessaires et suffisantes pour qu'elle soit $O(n^{3/2})$ avec une probabilité arbitrairement grande.*

EINLEITUNG

In [6] wurde eine Klasse $\mathscr{F}_Q$ von Abbildungen axiomatisch beschrieben, die zur redundanzfreien Codierung von Nachrichten einer diskreten Informationsquelle $Q=(\Sigma, p)$ verwendet werden sollen. Die zugrundeliegende Idee ist, grob gesprochen, die Nachrichten der Informationsquelle durch eine redundanzfreie Codierung so zu verschlüsseln, dass sie als Ausgabe einer gedächtnislosen Quelle mit identischen Buchstabenwahrscheinlichkeiten erscheinen und dadurch gegen statistische Verfahren der Kryptanalyse immun sind. Ein entscheidender Punkt für die Realisierbarkeit dieses Vorschlags ist die Komplexität dieser Coderungen:

$$f_{\tau,\pi}: \Sigma^* \to [0, 1] \subseteq \mathbb{R},$$

mit $f_{\tau,\pi} \in \mathscr{F}_Q$ bei geeigneter Zahlendarstellung.

(*) Received october 1984.

(**) Charakterierung redundanzfreier Codes zur Datenverschlüsselung, *R.A.I.R.O.*, Informatique théorique, Bd. 18, Nr. 2, 1984.

([1]) Department of Computer Science, The University of Western Ontario, London, Ontario, Canada, N6A 5B7.

([2]) Department of Mathematics, University of Arkansas, Fayetteville, AR, 72701, U.S.A.

Informatique théorique et Applications/Theoretical Informatics and Applications
ISSN en cours 86/01 5 25/$ 4.50/

Jelinek [4], Pasco [7] und Rubin [8, 9] diskutieren diese Codes für gedächtnislose Quellen und geben Algorihmen zu einer etwas modifizierten Codierung an: Um die Codeworte effizient zu berechnen, wird bei der Zahlendarstellung mit Registern fester Länge gerechnet und, sobald nötig, abgerundet. Dadurch wird, gerechtfertigt durch die ohnehin nötige Approximation bei realen Quellen, bewusst auf die für die Codes gerade charakteristische Redundanzfreiheit teilweise verzichtet. Es bleibt jedoch die Frage:Gibt es akzeptable Bedingungen, unter denen bereits der durch die Definition der $f_{\tau,\pi}$ implizierte natürliche Algorithmus mit hoher Wahrscheinlichkeit effizient arbeitet? Eine positive Antwort auf diese Frage ist selbstverständlich erst dann für die Konstruktion guter Kryptocodes besonders interessant, wenn es gelingt, sie auf Quellen allgemeineren Typs auszudehnen. Einen kleinen Schritt in dieser Richtung kann die Untersuchung der Situation für Markoffquellen bedeuten; diese wird in Abschnitt 2 durchgeführt. Zuvor wollen wir uns allerdings mit Quellen ohne Gedächtnis befassen.

1. REDUNDANZFREIE CODES FÜR GEDÄCHTNISLOSE QUELLEN

Sei nun $Q=(\Sigma, p)$ eine gedächtnislose Quelle. also :

$$p(u_1 u_2 \ldots u_m)=p(u_1)p(u_2)\ldots p(u_m)$$

and als Parameter für die Codierung seien, wie üblich nur $\tau=\bar{0}$ und solche Abbildungen π zugelassen, die aus einer einmaligen Permutation des Alphabets $\Sigma=\{a_1, \ldots, a_n\}$ entstehen (vgl. [6]); das soll heissen: $\pi(w, i)=\pi(\square, i)=:\pi(i)$ für all $w\in\Sigma^*$ ($\square$ bezeichnet stets das leere Wort). Der Schlüsselraum hat dann immerhin noch die Mächtigkeit $n!$, und die Formeln aus [6] für die Codierung vereinfachen sich erheblich:

Codierung einzelner Buchstaben:

$$q_i:=q_\pi(i)=\begin{cases} 0 & \text{für } i=0, \\ \sum_{j=1}^{i} p(a_{\pi(j)}) & \text{sonst.} \end{cases}$$

Streckungsfaktoren:

$$m(\square)=1, \qquad m(wa)=m(w)p(a).$$

Codeworte :

$$f_\pi(w)=\begin{cases} 0 & \text{für } w=\square, \\ f_\pi(\hat{w})+m(\hat{w}).q_{i-1} & \text{sonst, wobei } w=\hat{w}a_{\pi(i)} \text{ und } a_{\pi(i)}\in\Sigma. \end{cases}$$

O.B.d.A. setzen wir $p(a)>0$ für alle $a\in\Sigma$ voraus. Dann gilt auch $p(w)=m(w)>0$ für alle $w\in\Sigma^*$, und es ist:

$$0=q_0<q_1<\ldots<q_n=1.$$

f_π leistet bezüglich der durch π gegebenen Permutation des Alphabets Σ eine alphabetische Sortierung von Σ^*.

Man beweist leicht mit Induktion:

LEMMA 1.1:

$$f_\pi(a_{\pi(i_1)}a_{\pi(i_2)}\ldots a_{\pi(i_l)})=q_{i_1-1}+m(a_{\pi(i_1)})\cdot(q_{i_2-1}+m(a_{\pi(i_2)})\cdot(\ldots\cdot q_{i_l-1})\ldots),$$

und für $u,\ w\in\Sigma^*$:

$$f_\pi(uw)=f_\pi(u)+m(u)\cdot f_\pi(w).$$

Bei der Decodierung eines Codewortes $f_\pi(uw)$ ist nach bereits entschlüsseltem $u\in\Sigma^*$ das Codewort

$$f_\pi(w)=\frac{f_\pi(uw)-f_\pi(u)}{m(u)}$$

bekannt. Ist dann $f_\pi(w)\neq 0$ und $q_{i-1}\leqq f_\pi(w)<q_i$, so ist $w=a_{\pi(i)}w'$ und die Decodierung von $uw=ua_{\pi(i)}w'$ ist mit

$$f_\pi(w')=\frac{f_\pi(w)-q_{i-1}}{m(a_{\pi(i)})},$$

fortzusetzen. Oder aber es ist $f_\pi(w)=0$, und die gesendete Nachricht ist bestimmt als Präfix der Länge $\geqq|u|$ von:

$$ua_{\pi(1)}a_{\pi(1)}a_{\pi(1)}\ldots$$

Die Fälle I, III, IV der allgemeinen Decodierungssituation (vgl. [6]) scheiden hier aus.

1.1. Aufwandsabschätzungen

Aus den genannten, arithmetisch sehr einfachen *Abbildungen* f_π gewinnen wir eine *Codierung*, indem wir eine effektive Zahlendarstellung wählen, in der die arithmetischen Grundoperationen besonders effizient durchführbar sind. Dazu fixieren wir eine Basis $b\geqq 2$ und betrachten eine gedächtnislose Quelle $Q=(\Sigma, p)$, deren Buchstabenwahrscheinlichkeiten $p(a)$ als k-stellige Festkom-

mazahlen zur Basis b dargestellt sind. Dann ist die Codewortlänge $l_b(f_\pi(w))$ ebenso wie die Länge $l_b(m(w))$ des Streckungsfaktors durch $k \cdot |w|$ beschränkt. Dabei ist die absolute Länge $l_b(x)$ einer als Festkommazahl zur Basis b dargestellten Zahl $x \in [0, 1]$ definiert als die Anzahl der Stellen hinter dem Komma, wobei Nullen am rechten Ende der Darstellung von x weggelassen werden.

Wir interessieren uns zunächst für den sich aus der Definition von f_π ergebenden natürlichen Algorithmus A_b zur Berechnung von f_π in Festkommadarstellung bezüglich der Basis b; A_b benutze nur Register fester Länge. Die Implementierung von A_b ist keineswegs trivial, da die als Codeworte auftretenden Texte, als Zahlen interpretiert, sehr schnell eine praktisch nicht mehr handhabbare Grössenordnung erreichen.

Der Zeitaufwand für die Codierung von $w \in \Sigma^*$ mit f_π ist, wie sich aus obigen Formeln unmittelbar ergibt, durch $O(|w|^2)$ beschränkt. Während in anderen Problemkreisen die Wahl der Basis b für die Zahlendarstellung oft unwesentlich sein mag, zeigt sich hier, dass der mittlere Zeitaufwand für die Codierung mit f_π entscheidend davon abhängt, wie die Wahrscheinlichkeitsverteilung p und die Basis b zusammenpassen:

SATZ 1.2: $Q = (\Sigma, p)$ mit $\Sigma = \{a_1, \ldots, a_n\}$ sei eine gedächtnislose diskrete Quelle, bei der die Wahrscheinlichkeiten $p(a)$ für alle $a \in \Sigma$ rationale Zahlen sind. π sei eine Permutation von $\{1, \ldots, n\}$, $b \geqq 2$ eine natürliche Zahl. Es sei vorausgesetzt, dass die Exponenten der Primfaktorzerlegung von b teilerfremd sind. Dann gilt für jedes $\varepsilon > 0$: Der Zeitaufwand zur Berechnung eines Codewortes $f_\pi(w)$ gemäss A_b ist mit einer Wahrscheinlichkeit grösser als $1-\varepsilon$ beschränkt durch $O(|w|^{3/2})$, wenn für ein $k \in \mathbb{N}$ die folgenden Bedingungen erfüllt sind:

(i) $$h_i := p(a_{\pi(i)}) \cdot b^k \in \mathbb{N} \qquad \text{für} \quad i = 1, \ldots, n$$

und

(ii) $$\prod_{i=1}^{n} h_i^{h_i} = b^z \quad \text{für ein } z \in \mathbb{N}_0 := \mathbb{N} \cup \{0\},$$

oder äquivalent mit (ii) stattdessen:

(ii′) $$H_b(Q) := -\sum_{i=1}^{n} p(a_i) \log_b p(a_i) = k - \frac{z}{b^k} \quad \text{für ein } z \in \mathbb{N}_0.$$

Andernfalls wächst der Zeitaufwand für A_b mit Wahrscheinlichkeit grösser als $1-\varepsilon$ quadratisch mit $|w|$.

Beweis: Die Forderung (i) besagt nur, dass die $p(a)$ für $a \in \Sigma$ sämtlich eine endliche Darstellung zur Basis b besitzen müssen. Dies muss trivialerweise gelten, damit A_b als Algorithmus überhaupt definiert ist. Die Äquivalenz von (ii) und (ii') ist klar; beide Formeln sind gleichbedeutend mit

$$\sum_{i=1}^{n} h_i \log_b h_i \in \mathbb{N}_0.$$

Der Aufwand für die Codierung hängt im wesentlichen von der Grösse der Streckungsfaktoren $m(.)$ ab, genauer, von ihrer *relativen Darstellungslänge*: Die relative Darstellungslänge $\lambda_b(x)$ einer Festkommazahl $x = \sum_{i \in \mathbb{Z}} x_i b^i \neq 0$ mit $x_i \in \{0, 1, \ldots, b-1\}$ sei definiert durch:

$$\lambda_b(x) := \max_{i \in \mathbb{Z}} \{ i \mid x_i \neq 0 \} - \min_{i \in \mathbb{Z}} \{ i \mid x_i \neq 0 \} + 1.$$

Es ist also $\lambda_b(x) = 1$ genau dann, wenn x vielfaches $z \cdot b^c$ einer Potenz von b ist, wobei $z \in \{1, 2, \ldots, b-1\}$.

Beim Codieren eines Wortes $w \in \Sigma^*$ ist der Streckungsfaktor zunächst $m(\square) = 1$; für jeden codierten Buchstaben a wird er dann mit der entsprechenden Wahrscheinlichkeit $p(a\Sigma^\omega)$ multipliziert. Ist h_i' die Häufigkeit von $a_{\pi(i)}$ in w, $i = 1, \ldots, n$, so hat also $m(w)$ den Wert:

$$\prod_{i=1}^{n} p(a_{\pi(i)})^{h_i'}.$$

Um Eigenschaften natürlicher Zahlen auszunutzen, lassen wir das Komma weg, d. h., wir betrachten:

$$\bar{m}(w) := m(w) \cdot b^{k|w|} = \prod_{i=1}^{n} h_i^{h_i'}.$$

Der Beweis des Satzes verläuft in groben Zügen etwa folgendermassen:

(*a*) Mit dem zentralen Grenzwertsatz gilt wegen (ii), dass h_i' mit hoher Wahrscheinlichkeit nicht stark von $|w| h_i b^{-k}$ abweicht. Dann hat aber $m(w)$ mit hoher Wahrscheinlichkeit eine sehr kurze relative Darstellungslänge, und daher lassen sich die Multiplikationen mit $m(w)$ mit hoher Wahrscheinlichkeit in kurzer Zeit erledigen.

(*b*) Wenn (ii) nicht gilt, hat $m(w)$ mit hoher Wahrscheinlichkeit eine grosse (und wachsende) relative Darstellungslänge. Mit hoher Wahrscheinlichkeit braucht man für die Multiplikation mit $m(w)$ eine mit $|w|$ linear wachsende Zeit.

In einzelnen beweisen wir zunächst (a): $\pi_1, \ldots, \pi_m$ seien die Primfaktoren, die in b vorkommen, und:

$$\pi_1^{r_1} \ldots \pi_m^{r_m}$$

sei die Primfaktorzerlegung von b. Ist $z \in \mathbb{N}$ eine Zahl, in der nur $\pi_1, \ldots, \pi_m$ als Primfaktoren vorkommen, so sei:

$$\text{prim}(z) = (t_1, \ldots, t_m)$$

durch :

$$z = \pi_1^{t_1} \ldots \pi_m^{t_m}$$

gegeben. Wir betrachten die durch die Quelle definierte Folge $X_1, X_2, \ldots$ von unabhängigen Zufallsvariablen in $\mathbb{R}^m$ mit:

$$p(X_i = (t_{i_1}, \ldots, t_{i_m})) = p(a_j \in \Sigma, \text{prim}(h_j) = (t_{i_1}, \ldots, t_{i_m}))$$

und die Folge $Y_1, Y_2, \ldots$ mit :

$$Y_j = \sum_{i=1}^{j} X_i.$$

Offenbar ist :

$$p(Y_j = (s_{j_1}, \ldots, s_{j_m})) = p(w \in \Sigma^j, \text{prim}(\bar{m}(w)) = (s_{j_1}, \ldots, s_{j_m})).$$

Wir setzen :

$$\lambda_b(Y_j) = \lambda_b(\pi_1^{s_{j_1}} \ldots \pi_m^{s_{j_m}}),$$

wenn $Y_j = (s_{j_1}, \ldots, s_{j_m})$ ist. Ferner sei $\mu = \mathbf{E} X_i$; μ ist nach Voraussetzung von i unabhängig und es gilt:

$$\mathbf{E} Y_j = j\mu.$$

Nach dem zentralen Grenzwertsatz ist dann für alle $\varepsilon > 0$ und eine von ε abhängige Konstante c_ε mit Wahrscheinlichkeit grösser als $1-\varepsilon$:

$$\| Y_j - j\mu \| \lesssim c_\varepsilon \sqrt{j}.$$

Dabei bezeichnet $\| \cdot \|$ die euklidische Norm in $\mathbb{R}^m$. Ist nun (ii) erfüllt, so stimmt die durch $\mathbb{R}\mu$ definierte Gerade mit der durch die Basispotenzen definierten Geraden $\mathbb{R}(r_1, \ldots, r_m)$ überein. In diesem Falle ist also der

Abstand von $\mathbf{E}\, Y_j$ von dieser Geraden $\lesssim c_\varepsilon \sqrt{j}$. Ist andererseits (ii) nicht erfüllt, so ist $\mathbb{R}\mu \neq \mathbb{R}(r_1, \ldots, r_m)$, und der Abstand zwischen Y_j und $\mathbb{R}\mu$ ist $\lesssim c_\varepsilon \sqrt{j}$. Folglich wächst der Abstand d_j von Y_j von $\mathbb{R}(r_1, \ldots, r_m)$ mit Wahrscheinlichkeit $1-\varepsilon$ linear für $j \to \infty$. Man überlegt sich leicht, dass es Konstante x_1, x_2 gibt, so dass:

$$x_1 d_j \leqq \lambda_b(Y_j) \leqq x_2 d_j$$

gilt. Damit ist zunächst bewiesen: Für alle $\varepsilon > 0$ gilt:

$$\lambda_b(Y_j) = O(\sqrt{j}),$$

mit Wahrscheinlichkeit $1-\varepsilon$, falls (ii) erfüllt ist, und andernfalls:

$$\lambda_b(Y_j) \approx cj,$$

für eine geeignete Konstante c.

Zum Beweis des Satzes ist nur noch zu untersuchen, wie die Grösse von $\lambda_b(Y_j)$ den Rechenaufwand des Algorithmus beeinflusst.

Im Codierungsalgorithmus hat man zunächst zur Codierung jeweils eines Buchstabens die Grösse $m(u) \cdot q_{i-1}$ zu bestimmen mit dem Aufwand:

$$O(\lambda_b(q_{i-1}) \cdot \lambda_b(m(u))) = O(\lambda_b(Y_j)),$$

wenn man $Y_j = \mathrm{prim}(\bar{m}(u))$ setzt, wobei u ein Anfangsstück des zu codierenden Wortes w ist. Danach ist dann eine Addition der Form $f_\pi(u) + m(u) q_{i-1}$ erforderlich, deren Aufwand sich aus $\lambda_b(m(u) q_{i-1})$ und den bei der Addition entstehenden Überträgen ergibt. Um diesen Aufwand abzuschätzen, verwenden wir das folgende Lemma, das man leicht beweist.

LEMMA 1.3: *$(\{0, 1, \ldots, b-1\}, p)$ sei eine gedächtnislose Quelle mit identischen Buchstabenwahrscheinlichkeiten $1/b$ and X sei die Zufallsvariable über $\mathbb{N}_0$, die durch die Nachrichten dieser Quelle gegeben ist. Die Wahrscheinlichkeit, dass für $c \in \mathbb{N}_0$ und $y \leqq b^c$ bei der Addition $y + X$ die $(c+j)-$te Stelle von rechts in X berührt wird, ist dann kleiner als $(b-1)/b^j$.*

Wegen der Axiome E 1-E 3 (siehe [6]) können wir das Lemma auf:

$$X = f_\pi(u) \cdot b^{l_b(f_\pi(u))}$$

anwenden. Mit Induktion gilt:

$$l_b(f_\pi(u)) \leqq l_b(m(u)) + k.$$

Die Abschätzung aus Lemma 1.3 gilt erst recht, wenn man X rechts um einige Nullen verlängert.

Man betrachte also :

$$f_\pi(u)\, b^{l_b(m(u))+k}.$$

Hierzu ist :

$$(m(u)\cdot q_{i-1})\cdot b^{l_b(m(u))+k} \leqq b^{\lambda_b(m(u))+2k}$$

zu addieren. Setzt man für jede Stelle den Aufwand 1 an, so ist der mittlere Additionsaufwand durch:

$$\lambda_b(m(u))+2k+(b-1)\sum_{j=1}^{\infty}\frac{j}{b^{j-1}} = O(\lambda_b(m(u)))$$

abzuschätzen. Als Schranke für den Gesamtaufwand ergibt sich also:

$$O(|w|\,.\,\lambda_b(m(w))).$$

Man beachte dabei, dass sich der Additionsaufwand nicht wesentlich verringern lässt, so dass als Gesamtaufwand für A_b auch tatsächlich die Grössenordnung $|w|\cdot\lambda_b(m(w))$ zu erwarten ist. □

Die Voraussetzung über die Primfaktorzerlegung von b dient nur zur Vereinfachung und schränkt die Aussage des Satzes nicht wesentlich ein. Die Beschränkung auf den »natürlichen« Algorithmus A_b in Satz 1.2 lässt sich ohne grosse Schwierigkeit abschwächen, so dass dieser Satz unter einer intuitiv vernünftigen Voraussetzung, die eigentlich nur den deus ex machina ausschliesst, auch eine von der Wahl des speziellen Algorithmus unabhängige Komplexitätsaussage ergibt, die man mit einem Flaschenhalsargument beweist.

SATZ 1.4: Die Voraussetzungen von 1.2 seien erfüllt, und A_b' sei ein Algorithmus zur Berechnung von f_π zur Basis b mit der Eigenschaft, dass für alle $u, v\in\Sigma^*$ bei der Berechnung von $f_\pi(uv)$ auch $f_\pi(u)$ berechnet wird. Dann hat A_b' der Grössenordnung nach mindestens denselben Zeitaufwand wie A_b.

Beweis: Zur Berechnung von $f_\pi(w)$ für $w\in\Sigma^*$ müssen die Werte $f_\pi(u)$ für die $|w|+1$ Präfixe u von w bestimmt werden. Jeder solche Wert ist von den übrigen an ungefähr so vielen Stellen verschieden, wie die relative Darstellungslänge des betreffenden Streckungsfaktors ausmacht. Um diese Stellen umzustellen ist mindestens ein Aufwand in der Grössenordnung dieser Darstellungslänge erforderlich. □

BEMERKUNGEN 1.5

(1) *Zu den Primfaktoren*: Als notwendige Forderung, um die Entropiebedingung (ii) in 1.2 zu erfüllen, ergibt sich sofort, dass die Primfaktoren, die in den $h_1, \ldots, h_n$ aufgehen, dieselben sein müssen wie die Primfaktoren der Basis b. Darüberhinaus müssen die Potenzen, mit denen die einzelnen Primfaktoren in $h_1, \ldots, h_n$ *bzw.* in b vorkommen, genau aufeinander abgestimmt sein. Zur Simulation einer real gegebenen Quelle eine Basis für die Zahlendarstellung beliebig zu wählen und die Buchstabenwahrscheinlichkeiten durch Festkommazahlen zu approximieren, läuft daher fast immer besonders weit am Ziel vorbei und ergibt ausgesprochen langsame Verfahren. Als natürliche Zahl z in der Darstellung der Entropie der Quelle als:

$$H_b(Q) = k - \frac{z}{b^k}$$

kommen nur solche Zahlen $\leqq k \cdot b^k$ in Frage, die durch sämtliche Primfaktoren der Basis b, die aber nicht quadratisch in b aufgehen, teilbar sind: π_j sei einfacher Primfaktor von b. Wegen (ii) gibt es i mit $\pi_j \mid h_i$. Für alle derartigen i kommt $\pi_j^{\pi_j}$ als Teiler von $h_i^{h_i}$ vor, und daher wird $b^z = \prod_{i=1}^{n} h_i^{h_i}$ von $\pi_j^{\pi_j}$ geteilt; dabei hat π_j in b^z den Exponenten z. Also ist z Vielfaches von π_j.

(2) *Zeitaufwand, Spezialfälle:* Wie wir gesehen haben, wird der Aufwand bei oben beschriebener Codierung eines Wortes $w \in \Sigma^*$ im wesentlichen durch die relative Darstellungslänge $\lambda_b(m(u))$ der Streckungsfaktoren für die Präfixe u von w bestimmt. Nur wenn $m \circ \lambda_b$ generell durch eine Konstante beschränkt ist, kann der Codierungsaufwand auch im ungünstigsten Fall linear sein. Die Darstellungslänge λ_b für die Streckungsfaktoren ist genau dann durch eine Konstante beschränkt, wenn:

$$\operatorname{prim}(\bar{m}(\Sigma^*)) \subseteq \mathbb{R}(r_1, \ldots, r_m) = \mathbb{R}\operatorname{prim}(b),$$

ist; sobald nämlich ein einziger Punkt $\operatorname{prim}(\bar{m}(w)) \in \operatorname{prim}(\bar{m}(\Sigma^*))$ ausserhalb der durch die Basis b bestimmten Geraden $\mathbb{R}(\operatorname{prim}(b))$ liegt, erhält man für die Potenzen $m(w)^s = m(w^s)$, $s \in \mathbb{N}$, beliebig grosse Darstellungslängen $\lambda_b(m(w^s))$. Normiert man die Basis b wieder so, dass die Exponenten ihrer Primfaktoren insgesamt teilerfremd sind, ist also der Zeitaufwand für die Codierung im ungünstigsten Fall höchstens dann linear, wenn sämtliche Buchstabenwahrscheinlichkeiten der Quelle Potenzen der Basis b sind. Dieser Fall ist dadurch ausgezeichnet, dass alle Multiplikationen des Codierungsverfahrens auf Schiebeoperationen hinauslaufen, weil ja alle

Streckungsfaktoren Basispotenzen sind. Codeworte entstehen dann durch Addition der gegeneinander versetzten codierten Buchstaben. Für den Aufwand im ungünstigsten Fall sind die Überträge zu beachten, die zwar beliebig weit nach vorn laufen können, aber »dabei Nullen hinterlassen, so dass kein zweiter Übertrag später die Stelle passieren kann«. Für geeignete Permutationen π von Σ erhalten wir hier:

$$l_b(f_\pi(a_{\pi(i)})) = l_b(q_{i-1}) \leqq l_b(m(a_{\pi(i)}))$$

und damit durch Induktion:

$$l_b(f_\pi(wa_{\pi(i)})) = \max(l_b(f_\pi(w)),\ l_b(m(w)\cdot q_{i-1})) = l_b(m(w)) + l_b(q_{i-1})$$
$$\leqq l_b(m(w)) + l_b(m(a_{\pi(i)})) = l_b(m(wa_{\pi(i)})).$$

Genau für diese Permutationen π ergeben sich also die klassischen Huffman-Codes (mit Basispotenzen als Buchstabenwahrscheinlichkeiten).

(3) *Fehlerkorrektur und Basis* 2: Obwohl sich die beschriebene Codierung selbst nicht zur Fehlererkennung eignet, da die Redundanz der Nachrichten gerade beseitigt wird, lässt sich natürlich durch Kombination mit bekannten fehlerkorrigierenden Codes künstlich Redundanz zufügen, ohne die einmal erreichte Sicherheit der Kryptocodierung zu beeinflussen. Ein gewisses Mass an Redundanz entsteht ohnehin, wenn man aus technischen Gründen die Ziffern zur Basis b durch Dualzahlen $\leqq 2^{\lceil \log_2 b \rceil}$ darstellt und dadurch sowieso $2^{\lceil \log_2 b \rceil} - b$ Plätze frei behält.

(4) *Basisgrösse:* Um den Aufwand von A_b bei Variation von b zu vergleichen, sollte man ihn durch Multiplikation mit $O(\log b)$ normieren. Man beachte, dass sich b als k-te Wurzel für maximales k des Hauptnenners der p_i ergibt.

Beispiel 1.6: Häufigkeitsverteilungen mit erfüllter Entropiebedingung:

– zur Basis $b = 10$

Buchstabe	a_1	a_2	a_3-a_4	a_5	a_6-a_7	a_8	a_9-a_{12}	a_{13}-a_{17}	Summe
Haufigkeit h_i	25	20	*je* 10	8	*je* 5	4	*je* 2	*je* 1	100

– zur Basis $b = 60$

Buchstabe	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8-a_{12}	Summe
Häufigkeit h_i	15	12	10	8	5	3	2	*je* 1	60

1.2. Die Existenz schneller Codierungen

Der Satz 1.2 gibt Auskunft darüber, ob eine für die Zahlendarstellung vorgeschlagene Basis schnelles redundanzfreies Codieren gestattet oder nicht. Da die Anforderungen an die Basis, wie sich gezeigt hat, sehr scharf sind, verschiebt sich das Problem natürlich: Unter welchen Bedingungen existiert überhaupt eine passende Basis? Wie lässt sich gegebenfalls eine solche finden? Die basisabhängige Formulierung des Satzes 1.2 hilft hier nicht viel weiter; dennoch ist die Situation recht einfach zu beschreiben:

SATZ 1.7: Für eine gedächtnislose Quelle Q mit normierten Buchstabenhäufigkeiten $h_1, \ldots, h_n \in \mathbb{N}$, $n \geqq 2$, $\mathrm{ggT}(h_1, \ldots, h_n) = 1$, existiert genau dann eine Basis b für die Zahlendarstellung mit den Eigenschaften (i), (ii) aus Satz 1.2, wenn als Primfaktoren der $h_1, \ldots, h_n$ nur solche Primzahlen vorkommen, die auch in $\sigma := \sum_{i=1}^{n} h_i$ aufgehen und wenn darüberhinaus für die Vektoren $(\sigma_1, \ldots, \sigma_m)$, $(\rho_1, \ldots, \rho_m)$ der Primfaktorzerlegung von:

$$\sigma = \pi_1^{\sigma_1} \ldots \pi_m^{\sigma_m}$$

und:

$$\rho := \prod_{i=1}^{n} h_i^{h_i} = \pi_1^{\rho_1} \ldots \pi_m^{\rho_m}$$

gilt:

$$\sigma\sigma_i > \rho_i \qquad \text{für} \quad i = 1, \ldots, m.$$

Die Basis ist dann bis auf Potenzieren eindeutig bestimmt, d. h. es gibt eine Gerade $\mathbb{Q}(b_1, \ldots, b_m)$, so dass der Zeitaufwand im Sinne von Satz 1.2 zum Codieren mit $A_{b'}$ genau dann durch $O(|w|^{3/2})$ beschränkt ist, wenn prim $(b') \in \mathbb{N}^m \cap \mathbb{Q}(b_1, \ldots, b_m)$ ist.

Beispiele 1.8 für $\sigma = 20$.

Normierte Buchstabenhäufigkeiten	ρ	Geeignete Basis für die Codierung
16, 2, 1, 1	2^{66}	Existiert nicht
8, 8, 2, 1, 1	2^{50}	Existiert nicht
4, 4, 4, 4, 2, 1, 1	2^{34}	$78\,125\,000 = 2^3 \cdot 5^{10}$
5, 8, 4, 2, 1	$2^{34} \cdot 5^5$	$12\,500 = 2^2 \cdot 5^5$
5, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1	$2^{10} \cdot 5^5$	20
5, 5, 4, 2, 1, 1, 1, 1	$2^{10} \cdot 5^5$	40
5, 5, 4, 4, 2	$2^{18} \cdot 5^{10}$	$6\,400\,000 = 2^{11} \cdot 5^5$
5, 5, 8, 2	$2^{26} \cdot 5^{10}$	400 000
5, 5, 5, 2, 2, 1	$2^4 \cdot 5^{15}$	$2^{36} \cdot 5^5 = 214\,748\,364\,800\,000$ $\approx 2 \cdot 10^{14} \approx 2^{47}$
10, 5, 4, 1	$2^{18} \cdot 5^{15}$	$2^{22} \cdot 5^5$

Für $\sigma=6$ mit Buchstabenhäufigkeiten $h_1=3$, $h_2=2$, $h_3=1$ (vgl. [6], Beispiel 6) wächst die Länge der Streckungsfaktoren bei Basis 72 bereits erheblich langsamer als bei Basen 6 und 12. Optimale Basis ist jedoch 432.

Beweis von Satz 1.7: Da wir zur Diskussion der Existenz einer geeigneten Basis für die Zahlendarstellung die Primfaktorzerlegung natürlicher Zahlen benutzen wollen, beschreiben wir die gedächtnislose Quelle $Q=(\Sigma, p)$ mit rationaler Wahrscheinkeitsverteilung $p=(p(a_1), \ldots, p(a_n))$ durch Buchstabenhäufigkeiten $h_1, \ldots, h_n \in \mathbb{N}$. Es ist $p(a_i)=h_i/\sigma$ mit $\sigma := \sum_{i=1}^{n} h_i$. Diese Beschreibung durch Häufigkeiten ist natürlich nur bis auf einen gemeinsamen Faktor $x \in \mathbb{N}$ eindeutig bestimmt: Die sämtlichen Darstellungen von Q durch Buchstabenhäufigkeiten sind von der Form $(x \cdot h_1, \ldots, x \cdot h_n)$, falls $(h_1, \ldots, h_n) \in \mathbb{N}^n$ die durch $\operatorname{ggT}(h_1, \ldots, h_n)=1$ normierte Darstellung ist. Über den Aufwand zur Codierung bei Verwendung einer Basis $\beta=\sqrt[k]{x \cdot \sigma}$ entscheidet nach Satz 1.2 das Produkt:

$$\rho(x) := \prod_{i=1}^{n} (x \cdot h_i)^{x \cdot h_i} = (x^{\sigma} \cdot \rho)^x,$$

wobei:

$$\rho = \prod_{i=1}^{n} h_i^{h_i}.$$

Für $\mu \in \mathbb{N}$ sei $\boldsymbol{\mu}$ der Vektor $(\mu_1, \mu_2, \ldots)$ mit $\mu = p_1^{\mu_1} p_2^{\mu_2} \ldots$ wobei p_i die i-te Primzahl ist, $i=1, 2, \ldots$

Genau dann, wenn $\rho(x)$ eine Potenz von $\sqrt[k]{x \cdot \sigma}$ ist, ist der Aufwand für die Codierung durch $O(|w|^{3/2})$ beschränkt. Für die Vektoren der Primfaktorzerlegung lautet die äquivalente Bedingung:

$$\boldsymbol{\rho}(x) \in \mathbb{Q}(\sigma \mathbf{x}),$$

d. h.,

$$\boldsymbol{\rho} + \sigma \mathbf{x} = \lambda(\boldsymbol{\sigma} + \mathbf{x}),$$

mit $\lambda \in \mathbb{Q}$.

Es ergibt sich also das Gleichungssystem:

$$\rho_i + \sigma \cdot x_i = \lambda(\sigma_i + x_i), \qquad i=1, 2, 3, \ldots, \ldots \qquad (*)$$

Wir haben uns für die Lösungen $x \in \mathbb{N}$, $\lambda \in \mathbb{Q}$ dieses Systems (*) zu interessieren. Wir übergehen den Sonderfall, dass alle Häufigkeiten $h_1 = \ldots = h_n = 1$ sind. Für diesen Fall, in dem $\boldsymbol{\rho}$ der Nullvektor ist, zeigt man leicht, dass die Gerade $\mathbb{Q}\boldsymbol{\sigma}$ die Behauptung des Satzes erfüllt. Weiterhin ergibt sich $x_i = 0$ für den Fall $\rho_i = \sigma_i = 0$ als eindeutig bestimmte i-te Koordinate für die Lösung von (*). Es bleibt somit als Hauptfall folgendes endliche Gleichenssystem (**) zu betrachten:

$$\rho_i + \sigma \cdot x_i = \lambda(\sigma_i + x_i), \qquad i = 1, 2, \ldots, r, \tag{**}$$

wobei wenigstens ein $\rho_i \neq 0$ ist und $p_1, \ldots, p_r$ die Primzahlen sind, die in σ oder ρ aufgehen, und $\sigma = p_1^{\sigma_1} \ldots p_r^{\sigma_r}$, $\rho = p_1^{\rho_1} \ldots p_r^{\rho_r}$. Dann muss auch $\lambda \neq 0$ sein, und für jede Lösung von (**) ist $\rho_i + \sigma \cdot x_i = \lambda\,(\sigma_i + x_i) \in \mathbb{N}$, $i = 1, \ldots, r$; insbesondere auch $\sigma_i + x_i \neq 0$. Substituieren wir $y_i := \sigma_i + x_i$, so erhalten wir als äquivalente Formulierungen des Gleichungssystems (**) :

$$\rho_i + \sigma(y_i - \sigma_i) = \lambda y_i, \qquad i = 1, \ldots, r, \tag{3*}$$

$$\frac{\rho_i + \sigma(y_i - \sigma_i)}{\rho_j + \sigma(y_j - \sigma_j)} = \frac{y_i}{y_j}, \qquad 1 \leqq i, j \leqq r. \tag{4*}$$

Aus (4*) folgt (3*) mit :

$$\lambda = \frac{\rho_1 + \sigma(y_1 - \sigma_1)}{y_1} = \ldots = \frac{\rho_r + \sigma(y_r - \sigma_r)}{y_r},$$

$$(\rho_i - \sigma\sigma_i) y_j = (\rho_j - \sigma\sigma_j) y_i, \qquad 1 \leqq i, j \leqq r. \tag{5*}$$

Man kann sich natürlich bei (4*) und (5*) auf die Gleichungen mit $i = 1$ beschränken, da sich die übrigen als Konsequenzen ergeben. Wir kommen zu den notwendigen Bedingungen für die Lösbarkeit:

Das System (5*) ist nur lösbar, wenn alle $\sigma_i \neq 0$ sind, $i = 1, \ldots, r$. Annahme: $\sigma_j = 0$ für ein $j \leqq r$. Dann ist $\rho_j \neq 0$ nach Voraussetzung und es gilt für all Lösungen $\mathbf{y} \in \mathbb{N}^r$:

$$(\rho_i - \sigma\sigma_i) y_j = \rho_j y_i \in \mathbb{N} \qquad \text{für} \quad i = 1, \ldots, r.$$

Es folgt $\rho_i > \sigma\sigma_i$ und damit:

$$\rho = \prod_{i=1}^{r} p_i^{\rho_i} \geqq \prod_{i=1}^{r} p_i^{\sigma\sigma_i} = (\prod_{i=1} p_i^{\sigma_i})^{\sigma} = \sigma^{\sigma} \tag{+}$$

im Widerspruch zu $\rho = h_1^{h_1} \ldots h_n^{h_n} < \sigma^{h_1} \ldots \sigma^{h_n} = \sigma^{\sigma}$, da $n \geqq 2$.

Daraus, dass (+) nicht gelten kann, folgt, dass es stets ein $j \leqq r$ gibt mit $\sigma\sigma_j > \rho_j$. Darüberhinaus gilt:

Das System (5*) ist nur lösbar, wenn $\sigma\sigma_i > \rho_i$ für alle $i = 1, \ldots, r$. Denn ist $\rho_i \geqq \sigma\sigma_i$ und $\sigma\sigma_j > \rho_j$, so ist für alle $\mathbf{y} \in \mathbb{N}^r$:

$$(\rho_i - \sigma\sigma_i) y_j + (-(\rho_j - \sigma\sigma_j) y_i) > 0,$$

und daher eine Lösung der (i, j)-ten Gleichung aus (5*) unmöglich. Damit sind die Bedingungen aus Satz 1.7 als notwendig erkannt. Wir setzen nun $\sigma\sigma_i > \rho_i$ für $i = 1, \ldots, r$ voraus und betrachten die Lösungsmenge des dann zu (5*) äquivalenten Systems:

$$y_j = \frac{\sigma\sigma_j - \rho_j}{\sigma\sigma_1 - \rho_1} y_1, \qquad j = 2, 3, \ldots, r, \tag{6*}$$

wobei nur Lösungen $\mathbf{y} \in \mathbb{N}^r$ mit $y_i \geqq \sigma_i$ interessieren, $i = 1, \ldots, r$. Diese liefern uns dann die gewünschte Basis $b = p_1^{y_1} \ldots p_r^{y_r}$ für die Darstellung der Codeworte. Als Lösungskomponente y_1 von (6*) eignen sich gerade die Vielfachen von:

$$\frac{\sigma\sigma_1 - \rho_1}{\operatorname{ggT}\{\sigma\sigma_i - \rho_i \mid 1 \leqq i \leqq r\}},$$

und die gesuchte Gerade des $\mathbb{Q}^r$ ist der Durchschnitt der durch (6*) gegebenen $r-1$ Hyperebenen, nämlich:

$$\mathbb{Q}(\sigma\sigma_1 - \rho_1, \ldots, \sigma\sigma_r - \rho_r). \quad \square$$

1.3. Übersicht zur Effizienz redundanzfreier Codierung

Wie wir gesehen haben, gibt es einerseits solche gedächtnislose Quellen mit rationalen Buchstabenwahrscheinlichkeiten, für die redundanzfreie Codierung quadratischen Zeitaufwand erfordert und somit recht wenig attraktiv ist, und andererseits solche, bei denen sich die Primfaktoren der Basis so aufeinander abstimmen lassen, dass man mit einen Zeitaufwand $O(|w|^{3/2})$ für die Codierung auskommt, wobei dann die geeignete Basis für die Zahlendarstellung im wesentlichen eindeutig festliegt. Genauer besehen, ist das jedoch nur die halbe Wahrheit:

Es gibt durchaus Situationen in denen der asymptotisch wesentlich günstigere Zeitaufwand $O(|w|^{3/2})$ nur so teuer zu erkaufen ist, dass er praktisch nutzlos ist. Dazu ein kleines Beispiel:

Beispiel 1.9:

Buchstabenhäufigkeiten : 25, 25, 16, 8, 8, 5, 5, 5, 2, 1;

Summe σ : 100;

Produkt ρ : $2^{114}5^{115}$;

Kleinste Basis für wahrscheinlichen Zeitaufwand $O(|w|^{3/2})$: $2 \cdot 10^{85}$;

Effizienzvergleich: Die zu erwartende relative Darstellungslänge λ der Streckungsfaktoren beträgt nach z codierten Buchstaben etwa:

Buchstabenanzahl z Massstab für λ	100		1 000		10 000		10^5		10^6	
	A	B	A	B	A	B	A	B	A	B
Basis 2	Unmöglich									
Basis 10	1	3	7	23	70	230	700	2 300	7 000	23 000
Basis $2 \cdot 10^{85}$	86	290	86	290	86	290	86	290	36	290
A, dezimal; B, dual.										

Dieses ist ein ausgesprochen charakteristisches Beispiel: Die Situation lässt uns die Wahl zwischen schliesslich extrem ansteigendem Aufwand einerseits und relativ günstigem Aufwand pro Buchstaben andererseits, der aber durch gigantischen Aufwand für kurze Wörter erkauft wird. Man lernt, dass schnelle Algorithmen in der Regel nur solange effizient sind, als sie in Grössenordnungen eingesetzt werden, für die sie konzipiert sind. Im Gegensatz zum früher studierten Minimalbeispiel mit $\sigma = 6$ ist hier die kleinstmögliche Basis 10 für kurze Worte sehr gut; dafür lässt sich jedoch das Verfahren hier nicht für den mittleren Bereich (hierbei etwa 10^4) optimieren. Bei $\sigma = 6$ stellte die Basis 72 einen Kompromiss dar. Auf die Optimierung der Basis für gewisse Bereiche gehen wir nicht näher ein; die Vektoren der Primfaktorzerlegung geben eine anschauliche Vorstellung davon.

Wie wir gesehen haben, kann man unter Umständen schon bei gedächtnislosen Quellen in ziemlich ausweglose Situationen geraten. Bei Quellen mit Gedächtnis wird die Lage naturgemäss komplizierter, und es ist zu erwarten, dass redundanzfreie Codierung nur umso aufwendiger wird. Es wird sich jedoch zeigen, dass zumindest für Markoffquellen die Situation nur unwesentlich komplizierter ist.

2. REDUNDANZFREIE CODES FÜR MARKOFFQUELLEN

Um den Verhältnissen in der Praxis einen Schritt näher zu kommen, betrachten wir nun Informationsquellen mit Gedächtnis. Formal werden wir solche Quellen diskutieren, die die Markoff-Eigenschaft haben, so dass die Wahrscheinlichkeit eines gesendeten Nachrichtensymbols nur vom vorange-

henden abhängt. Der für Anwendungen entscheidende Punkt jedoch ist die sorgfältige Auswahl der Nachrichtensymbole: Es ist nämlich nur als Trivialfall beabsichtigt, Codes für Sprachen zu studieren, bei denen die bedingte Häufigkeit eines Buchstaben nur vom letzten vorangehenden Buchstaben abhängt. In Wirklichkeit benötigen wir als Gedächtnis eine endliche, geeignet strukturierte Zustandsmenge, die Aufgaben aus unterschiedlichsten Bereichen übernehmen kann. Da diesbezügliche Einzelheiten über den Rahmen der Arbeit hinausgehen, begnügen wir uns mit einigen Stichworten, die Prinzipielles erkennen lassen und dadurch den theoretischen Ansatz motivieren. Dass Sorgfalt bei der Auswahl der Zustände geboten ist, zeigt schon folgendes triviale Beispiel: Bei einem Alphabet mit 26 Buchstaben erfordert ein Schieberegister der Länge 5 mehrere Millionen Zustände, nämlich genau 26^5. Trotzdem ist der Nutzen gering, da sich mit einem solchen Gedächtnis der Länge 5 allenfalls der Silbenbau der Sprache berücksichtigen lässt.

Wünschenswert dagegen ist es, ausser einem kurzen, reduzierten Schieberegister für den Silbenbau noch ein Gedächtnis über unbeschränkte Zeit in die Zustandsmenge einzubauen, um z. B. in gewissen Textabschnitten häufig vorkommende Eigennamen zu berücksichtigen, die dann vorher zu spezifizieren sind (»programmierbare Codes«), und evtl. die Beschränkung auf ein Thema oder eine Teilsprache zu signalisieren, wie z. B. 'Control Tower Language' [3]. Denn es ist klar, dass die zur redundanzfreien Codierung benötigte Sprachstatistik die Statistik der aktuell zu codierenden Nachrichten sein muss und nicht etwa die Statistik irgendeiner Obersprache (extremes Beispiel: Sprache = Deutsch, zu sendende Nachrichten = Telefonverzeichnisse).

Sendet man nun die verschlüsselten Buchstaben einer Sprache, ohne die Zustände der Quelle mitzuteilen, die ja die Codierung wesentlich beeinflussen, so ergeben sich für den Decodierer Probleme. Denn im allgemeinen wird sich aus den bereits gesendeten Buchstaben der Zustand der Quelle, der für die Verschlüsselung des folgenden Buchstaben massgeblich ist, entweder gar nicht oder noch nicht eindeutig ermitteln lassen (»Decodierungsverzögerung«). Dazu vergleiche man die »unifilar Markov source« bei Ash [1] und die »state-calculable Markov source« bei Jelinek und Schneider [5]. Es erscheint daher zweckmässig, überhaupt nicht die Worte der Sprache als Nachrichten zu codieren, sondern vielmehr die Folge der Zustände, die die Quelle nacheinander annimmt und die sich ihrerseits leicht so konzipieren lassen, dass sie eindeutig das zu übermittelnde Wort der Sprache bestimmen. Die Information, die eine Quelle als Nachricht senden kann, ist durch die Folge der Zustände eindeutig festgelegt, während die gesendeten Worte durchaus mehrdeutig sein können und – ähnlich wie in Alltagssituationen – oft keinen

eindeutigen Rückschluss auf den Zustand der Quelle (z. B. die Absicht des Sprechers) erlauben.

Sei also nun eine diskrete Markoffquelle $Q=(S, P)$ mit Zustandsmenge $S=\{s_1, s_2, \ldots, s_n\}$ und Zustandsübergangsmatrix:

$$P=\begin{pmatrix} p(s_1|s_1) \ldots p(s_1|s_n) \\ \vdots \\ p(s_n|s_1) \ldots p(s_n|s_n) \end{pmatrix},$$

gegeben. $p(s_i|s_j)$ ist die Wahrscheinlichkeit, mit der Q von s_j im nächsten Schritt in den Zustand s_i übergeht. Eine etwaige Symbolausgabefunktion ist für die zu studierende redundanzfreie Codierung von Q unwesentlich. Für Codierung und Decodierung möchten wir im Interesse der Effizienz möglichst einfache Rekursionsformeln angeben, die etwa mit denen für gedächtnislose Quellen zu vergleichen sind. Es wird sich als sinnvoll erweisen, den Startzustand s_{j_0} der Quelle Q offen zu halten und gleichzeitig n Codierungen:

$$f_\pi : S^* \times \{1 \ldots, n\} \to [0, 1] \subseteq \mathbb{R}$$

zu betrachten, so dass $f_\pi(w|j_0)$ das Codewort für $w \in S^*$ ist, wenn die Quelle Q im Zustand s_{j_0} gestartet wird. Analog zum Fall der gedächtnislosen Quelle setzen wir ohne wesentliche Beschränkung der Allgemeinheit immer den Parameter $\tau=\bar{0}$ voraus, und für die Permutationen π_w von $\{1, \ldots, n\}$ fordern wir, motiviert durch die Markoffeigenschaft:

$$\pi(\square, i)=\pi(s_{j_0}, i)$$

für den Code bei Startzustand s_{j_0} und:

$$\pi(ws_j, i)=\pi(s_j, i)=:\pi(i|j)$$

für alle $w \in S^*$. Dabei sind also $\pi(s_{j_0}, .)$ und $\pi(.|j)$ Permutationen von $\{1, \ldots, n\}$. Die Codierung f_π bei Startzustand s_{j_0} wird dann beschrieben durch:

Codierungen einzelner Buchstaben:

$$q_{0j}=0,$$

$$q_{ij}=\sum_{k=1}^{i} p(s_{\pi(k|j)}|s_j).$$

Streckungsfaktoren:

$$m(s_i \mid j_0) = p(s_i \mid s_{j_0}),$$
$$m(ws_j s_i \mid j_0) = m(ws_j \mid j_0) \cdot p(s_i \mid s_j).$$

Codierung von Wörtern $\in S^+$ *:*

$$f_\pi(s_{\pi(i \mid j_0)} \mid j_0) = q_{i-1,\, j_0},$$
$$f_\pi(ws_j s_{\pi(i \mid j)} \mid j_0) = f_\pi(ws_j \mid j_0) + m(ws_j \mid j_0) \cdot q_{i-1,\, j}.$$

Zerlegung der Codewörter für die Decodierung :

$$f_\pi(us_j w \mid j_0) = f_\pi(us_j \mid j_0) + m(us_j \mid j_0) \cdot f_\pi(w \mid j).$$

Einsetzen von $u = \square$ und Auflösen der letzten Gleichung nach $f_\pi(w \mid j)$ ergibt die für die Decodierung benötigte Rekursionsvorschrift. Dabei werden die Codierungen für Startzustände $s_j \neq s_{j_0}$ benötigt; ansonsten verläuft das Verfahren wie bei gedächtnislosen Quellen. Um wieder ein Kriterium dafür zu gewinnen, wann die oben definierte Codierung praktikabel ist, betrachten wir die Äquivalenzrelation $\vdash\!\dashv$ auf S, die von der Erreichbarkeisrelation:

$$s_j \vdash s_i \quad \Leftrightarrow i = j \text{ oder es gibt } i_0, i_1, \ldots, i_r$$
$$\text{mit } j = i_0, \qquad i = i_r, \qquad p_{i_k i_{k+1}} > 0 \text{ für } k = 0, 1, \ldots, r-1,$$

erzeugt wird. Die durch $\vdash$ auf $S/\vdash\!\dashv$ induzierte partielle Ordnung bezeichnen wir wieder mit $\vdash$. Da wir uns für die Codierung bei Startzustand s_{j0} interessieren, wollen wir voraussetzen, dass alle Zustände von Q von s_{j_0} aus erreichbar sind. Die maximalen Elemente von $S/\vdash\!\dashv$ seien $S_1, S_2, \ldots, S_t$. Die Zustände $s \in S_1 \cup \ldots \cup S_t$ sind also die rekurrenten Zustände von Q; diese sind für die Codierung langer Wörter offenbar entscheidend. Die Zustände $s \in S \setminus \bigcup_{i=1}^{t} S_i$ sind transient; sie kommen mit hoher Wahrscheinlichkeit nur für ein Anfangsstück der Nachricht in Betracht, dessen Länge für die Praxis allerdings von grosser Bedeutung sein kann. Die durch $S_1, \ldots, S_t$ bestimmten Markoffquellen $Q_1 = (S_1, P_1), \ldots, Q_t = (S_t, P_t)$ sind nach Konstruktion unzerlegbar und haben daher eindeutig bestimmte stationäre Verteilungen:

$$C_1 = (p_{11}, \ldots, p_{1n}), \ldots, C_t = (p_{t1}, \ldots, p_{tn})$$

mit $p_{ij}=0$ für $s_j \in S \setminus S_i$. Der mittlere Zeitaufwand für die Codierung hängt dann unmittelbar ab von:

– dem Hauptnenner σ_0 der $p(s_i | s_j)$, die als rational vorauszusetzen sind;

– dem Hauptnenner σ_l der $p_{l,i}$ für $l=1, \ldots, t$ und

– dem analog zum gedächtnislosen Fall gebildeten Produkt ρ_l der Häufigkeiten, das C_l für $l=1, \ldots, t$ berücksichtigt. Genauer geben die Sätze 2.1 and 2.2 darüber Auskunft. Die Voraussetzungen und Bezeichnungen für diese Sätze sind:

$Q=(S, P)$ sei eine diskrete Markoffquelle mit Startzustand s_{j_0} und Matrix $P \in \mathbb{Q}^{n \times n}$ der Wahrscheinlichkeiten für die Zustandsübergänge. $Q_1=(S_1, P_1), \ldots, Q_t=(S_t, P_t)$ seien die schliesslich erreichbaren unzerlegbaren Komponenten mit den stationären Verteilungen:

$$C_1=(p_{11}, \ldots, p_{1n}), \ldots, C_t=(p_{t_1}, \ldots, p_{t_n}).$$

Diese sind rational. Für $l=1, \ldots, t$ sei σ_l der Hauptnenner von $p_{t_1}, \ldots, p_{t_n}$. A_b sei der natürliche Algorithmus zur Codierung $w \mapsto f_\pi(w | j_0)$ bei Zahlendarstellung zur Basis b. Ohne Beschränkung der Allgemeinheit sei vorausgesetzt, dass b keine Potenzzahl ist. Für eine beliebige $n \times n$-Matrix $M=(m_{ji})$ und $l=1, \ldots, t$ sei:

$$\rho_l(M) := \prod_{j=1}^{n} \left(\prod_{i=1}^{n} m_{ji}^{m_{ji}} \right)^{p_{lj}\sigma_l},$$

mit $0^0 := 1$. Dann gilt, sofern kein Q_l für $l=1, \ldots, t$ deterministisch ist [1]:

SATZ 2.1: Für jedes $\varepsilon > 0$ arbeitet der Algorithmus A_b mit Wahrscheinlichkeit $1-\varepsilon$ mit dem Zeitaufwand $O(|w|^{3/2})$, wenn für eine geeignete Basispotenz b^k gilt:

(i) $\qquad p(s_i | s_j) b^k \in \mathbb{N}^0 \qquad$ für $\quad i, j=1, \ldots, n$

und

(ii) $\qquad \rho_l(b^k P) \in b^{\mathbb{N}_0} \qquad$ für $\quad l=1, \ldots, t,$

oder, äquivalent mit (ii),

(ii') $\qquad H_b(Q | C_l) = k - \dfrac{z_l}{b^k \cdot \sigma_l} \qquad$ mit $\quad z_l \in \mathbb{N}_0.$

[1] Dies ist wegen Axiom E 4 erforderlich (vgl. [6]).

Anderenfalls gibt es $\alpha > 0$, so dass A_b mit einer Wahrscheinlichkeit $> \alpha$ quadratischen Aufwand hat für $|w| \to \infty$.

SATZ 2.2: Sei σ_0 der Hauptnenner der $p(s_i|s_j)$ und $\rho_l := \rho_l(\sigma P)$ für $l = 1, \ldots, t$. Es sei $b = \pi_1^{r_1} \ldots \pi_m^{r_m}$ die Primfaktorzerlegung von b und

$$\operatorname{prim} \sigma_0 = (\sigma_{01}, \ldots, \sigma_{0m}), \qquad \operatorname{prim} \rho_l = (\rho_{l1}, \ldots, \rho_{lm}).$$

A_b benötigt für die Codierung mit $f_\pi(w|j_0)$ mit beliebig grosser Wahrscheinlichkeit höchstens die Zeit $O(|w|^{3/2})$ genau dann, wenn:

$$\operatorname{prim} b = (r_1, \ldots, r_m) \in \prod_{l=1}^{t} \mathbb{Q}(\sigma_l \sigma_0 \operatorname{prim} \sigma_0 - \operatorname{prim} \rho_l)$$

gilt. Eine solche Basis b existiert genau dann, wenn:

$$\rho_{1i} < \sigma_1 \sigma_0 \sigma_{0i}$$

für $i = 1, \ldots, m$ ist und die Geraden:

$$\mathbb{Q}(\sigma_l \sigma_0 \operatorname{prim} \sigma_0 - \operatorname{prim} \rho_l)$$

für $l = 1, \ldots, t$ übereinstimmen.

Beweise: Der Beweis von Satz 2.1 verläuft zu grossen Teilen analog dem für gedächtnislose Quellen. Wir heben nur die Unterschiede hervor. Auch hier hängen Zeitaufwand und relative Darstellungslänge der Streckungsfaktoren eng zusammen: Ist die relative Darstellungslänge $\lambda_b(m(w|j_0))$ mit hoher Wahrscheinlichkeit durch $O(\sqrt{|w|})$ beschränkt, so bleibt der mittlere Aufwand zum Codieren durch $O(|w|^{3/2})$ beschränkt; wächst dagegen $\lambda_b(m(w|j_0))$ mit Wahrscheinlichkeit $> \alpha > 0$ linear mit $|w|$, so ist der Aufwand mit dieser Wahrscheinlichkeit von derselben Grössenordnung wie im ungünstigsten Fall, nämlich quadratisch mit $|w|$.

Es ist also wieder nur das Verhalten vom $\lambda_b(m(w|j_0))$ *zu* studieren. Ausgehend vom Startzustand s_{j_0} wird schliesslich mit Wahrscheinlichkeit 1 eine bezüglich $\vdash$ maximale Klasse von Zuständen erreicht, und nicht wieder verlassen. Dies sei etwa S_l. Da wir uns für das Grenzverhalten des Aufwands zur Berechnung von $f_\pi(w|j_0)$ interessieren, zerlegen wir:

$$w = u s_{jl} w',$$

so dass:

$$s_{jl} \in S_l \qquad \text{und} \qquad u \notin S^* S_l$$

ist. Dabei gilt dann :

$$f_\pi(w\,|\,j_0) = f_\pi(us_{jl}\,|\,j_0) + m(us_{jl}\,|\,j_0)\, f_\pi(w'\,|\,j_l).$$

Dabei können wir den Aufwand für diese Zusammensetzung von $f_\pi(w\,|\,j_0)$ aus $f_\pi(us_{jl}\,|\,j_0)$, $m(us_{jl}\,|\,j_0)$ und $f_\pi(w'\,|\,j_l)$ und den mit $|u|$ höchstens quadratisch wachsenden Aufwand für die Bestimmung von $f_\pi(us_{jl}\,|\,j_0)$ und $m(us_{jl}\,|\,j_0)$ im Hinblick auf die exponentiell fallende Wahrscheinlichkeit langer Folgen transienter Zustände durch eine Konstante abschätzen. Zur Abschätzung des Aufwandes zur Berechnung von $f_\pi(w'\,|\,j_l)$ brauchen wir nur noch die Quelle $Q_l = (S_l, P_l)$ zu betrachten. Die stationäre Verteilung C_l von Q_l gibt an, mit welchen Wahrscheinlichkeiten die Zustände bei längerem Laufen von Q_l angenommen werden. Ist w' sehr lang, so werden mit hoher Wahrscheinlichkeit zur Berechnung des Streckungsfaktors $m(w'\,|\,j_l)$ die Zahlen $p(s_i\,|\,s_j)$ entsprechend der Verteilung C_l miteinander multipliziert, nämlich jedes $p(s_i\,|\,s_j)$ mit einer zu $p(s_i\,|\,s_j)$ proportionalen Häufigkeit. Das erwartete Verhältnis der Exponenten der Primfaktoren von;

$$m(w'\,|\,j_l) \cdot b^{l_b(m(w'\,|\,j_l))}$$

zueinander ist also dasselbe wie in :

$$\rho_l(b^k P) = \prod_{j=1}^{n} \left(\prod_{i=1}^{n} (b^k p(s_i\,|\,s_j))^{b^k p(s_i\,|\,s_j)} \right)^{p_{lj}\sigma_l},$$

wobei die Abweichung für jedes $\varepsilon > 0$ wieder durch $O(\sqrt{|w'|})$ mit Wahrscheinlichkeit $1-\varepsilon$ beschränkt ist([1]).

Die restliche Argumentation zu 2.1 verläuft dann wie in Beweis 1.2. Bei dem Beweis des quadratischen Aufwands für den Fall, dass (ii) nicht erfüllt ist, etwa für $\rho_j(b^k P)$, folgt man für Q_j der Argumentation von 1.2 und beachtet, dass S_j mit einer Wahrscheinlichkeit $> \alpha_j > 0$ schliesslich erreicht wird.

Die Äquivalenz von (ii) und (ii') schliesslich ergibt sich folgendermassen:

$$\log_b \rho_l(b^k P) = \sum_{j=1}^{n} p_{lj}\sigma_l \left(\sum_{i=1}^{n} b^k p(s_i\,|\,s_j)\,(k + \log_b p(s_i\,|\,s_j)) \right)$$

$$= b^k \sigma_l \left(k + \sum_{j=1}^{n} p_{lj} \sum_{i=1}^{n} p(s_i\,|\,s_j) \log_b p(s_i\,|\,s_j) \right) = b^k \sigma_l (k + H_b(Q\,|\,C_l)).$$

([1]) Zum zentralen Grenzwertsatz für Markoffketten vgl. z. B. [2], I, § 16.

$\rho_l(b^k P)$ ist somit genau dann eine Potenz b^{z_l} mit $z_l \in \mathbb{N}_0$, wenn (ii') gilt.

Zum Beweis von Satz 2.2 überlegt man zunächst: Da eine Basis b, soll sie überhaupt für die Codierung in Frage kommen, mit einer Potenz b^k eine Beschreibung von Q durch Häufigkeiten $b^k p(s_i \mid s_j) \in \mathbb{N}_0$ leisten muss, ist offenbar σ_0 die kleinste Zahl, die als b^k möglich ist. Sämtliche weitere Möglichkeiten ergeben sich durch Multiplikation der:

$$h_{ji} := \sigma_0 p(s_i \mid s_j)$$

mit einem gemeinsamen Faktor $x \in \mathbb{N}_0$. Wir untersuchen, wie x den für den Zeitaufwand entscheidenden Ausdruck $\rho_l(x\,\sigma_0 P)$ beeinflusst:

$$\rho_l(x\,\sigma_0 P) = (x^{\sigma_0 \sigma_l}\,\rho_l(\sigma_0 P))^x = (x^{\sigma_0 \sigma_l}\,\rho_l)^x,$$

wobei man :

$$\sum_{i=1}^{n} h_{ji} = \sigma_0,$$

beachtet. Dass $\rho_l(x\,\sigma_0 P)$ eine Basispotenz ist, bedeutet, in den Vektoren der Primfaktorzerlegung ausgedrückt ([1]):

$$\boldsymbol{\rho}_l + \sigma_0\,\sigma_l\,\mathbf{x} = \mu_l(\boldsymbol{\sigma}_0 + \mathbf{x})$$

mit $\mu_l \in \mathbb{Q} \setminus \{0\}$. Die Substitution $\mathbf{y} := \boldsymbol{\sigma} + \mathbf{x}$ ergibt als Gleichungssystem:

$$\boldsymbol{\rho}_l + \sigma\sigma_l(\mathbf{y} - \boldsymbol{\sigma}) = \mu_l\,\mathbf{y},$$

dessen Lösungen y gerade den Primfaktorzerlegungen der geeigneten Basen entsprechen.

Zunächst zeigt man, dass die Menge:

$$L := \bigcap_{l=1}^{t} \mathbb{Q}(\sigma_l\,\sigma_0\,\boldsymbol{\sigma}_0 - \boldsymbol{\rho}_l) \setminus \{0\},$$

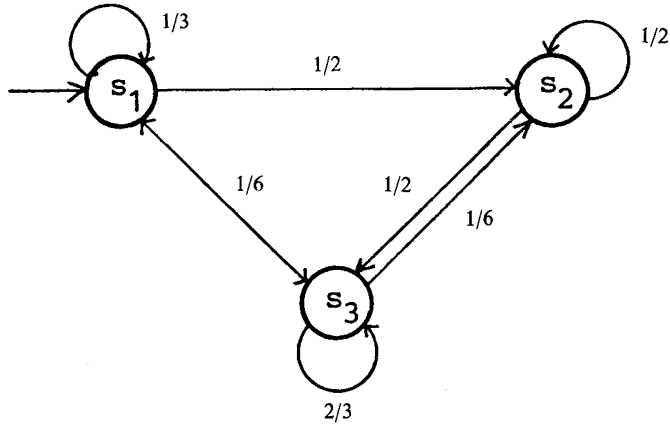
die Lösungsmenge dieses Gleichungssystems ist. Im zweiten Schritt zeigt man dann, dass eine Lösung existiert, d. h., $L \cap \mathbf{N} \neq \varnothing$ gilt genau dann, wenn die Geraden $\mathbb{Q}(\sigma_l\,\sigma_0\,\boldsymbol{\sigma}_0 - \boldsymbol{\rho}_l)$ für $l = 1, \ldots, t$ zusammenfallen und wenn:

$$\rho_{1i} < \sigma_1\,\sigma_0\,\sigma_{0i},$$

für $i = 1, \ldots, m$ ist. □

([1]) Bezeichnung wie im Beweis vom 1.7.

Beispiel 2.3: $Q=(S, P)$ sei die durch folgenden Zustandsgraphen gegebene Markoffquelle:



Matrix P | Stationäre Verteilung | Matrix der nomierten Häufigkeiten

$$\begin{pmatrix} \frac{1}{3} & 0 & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{2} & \frac{2}{3} \end{pmatrix} \quad \begin{pmatrix} \frac{3}{22} \\ \frac{7}{22} \\ \frac{12}{22} \end{pmatrix} \quad \begin{pmatrix} 2 & 0 & 1 \\ 3 & 3 & 1 \\ 1 & 3 & 4 \end{pmatrix}$$

$$\sigma_0=6, \quad \sigma_1=22, \quad \rho_1=2^{102}3^{51},$$

$$\mathbb{Q}(\sigma_1\ \sigma_0\ \boldsymbol{\sigma}_0-\boldsymbol{\rho}_1)=\mathbb{Q}(30, 81)=\mathbb{Q}(10, 27).$$

Kleinste geeignete Basis für Algorithmus A is also $2^{10}3^{27}$.

Vergleich mit den gedächtnislosen Quellen, die den einzelnen Zuständen s_1, s_2, s_3 entsprechen:

Zustand	Buchstabenhäufigkeiten	Gerade der geeigneten Basispotenzen
s_1	2, 3, 1	$\mathbb{Q}(4, 3)$ (Minimalbeispiel)
s_2	0, 3, 3 (äq. zu 0, 1, 1)	$\mathbb{Q}(1, 0)$
s_3	1, 1, 4	Existiert nicht

Eine Abänderung von Q im Zustand S_2 ergibt eine Quelle mit folgenden Daten:

Matrix P stationäre verteilung Häufigkeitsmatrix

$$\begin{pmatrix} \frac{1}{3} & \frac{1}{2} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{6} \\ \frac{1}{6} & 0 & \frac{2}{3} \end{pmatrix} \quad \begin{pmatrix} \frac{6}{16} \\ \frac{7}{16} \\ \frac{3}{16} \end{pmatrix} \quad \begin{pmatrix} 2 & 3 & 1 \\ 3 & 3 & 1 \\ 1 & 0 & 4 \end{pmatrix}$$

Kleinste geeignete Basis ist dann $2^5 \cdot 3^3 = 864$.

SCHLUSSBEMERKUNGEN

Mit den Überlegungen in [6] und in dieser Arbeit konnten wir:

– eine axiomatische und eine rekursive Charakterisierung einer natürlichen Klasse redundanzfreier Codes angeben;

– und für gedächtnislose Quellen und Markoffquellen den wahrscheinlichen Codierungsaufwand für die naheliegenden Spezialfälle abschätzen.

Diese Ergebnisse sind allgemeiner und präziser als das bisher Bekannte, weil sonst üblicherweise mit Rücksicht auf die algorithmischen Probleme mit approximierenden Verfahren gearbeitet wird. Eine nicht-triviale Abschätzung des Decodierungsaufwands steht noch aus.

In der Praxis werden die Codierungen $f_{\tau,\pi}$ kryptanalytisch verwundbar sein. Dies liegt nicht am Codierungsprinzip selbst, sondern daran, dass die jeweils betrachtete stochastische Quelle die Realität nur in mehr oder minder adäquater Weise beschreibt. Dass sich die für gedächtnislose Quellen konstruierten Codes nicht auf andere Quellen in der Praxis anwenden lassen, ist leicht zu sehen: Sind nämlich etwa die Anfangsstücke von gesendeten Nachrichten für eine »known-plaintext«-Attacke verfügbar, so lässt sich die als Schlüssel benutzte Permutation der Buchstaben je nach Umfang des bekannten Texts teilweise oder ganz ermitteln (vgl. [8]). Ähnlich lassen sich mit hoher Wahrscheinlichkeit auftretende Wörter ausnutzen. Sicher ist, dass endliche Markoffquellen besser geeignet sind als gedächtnislose Quellen. Aber auch sie beschreiben die mögliche Vielfalt der Wirklichkeit für kryptographische

Zwecke nur ungenügend. Besonderes Augenmerk verdient dabei die Variation der statistischen Charakteristiken einer Quelle, wie sie in der Realität in der Regel anzutreffen ist. Dieses im allgemeinen *a priori* schwer beschreibbare und durch Mittelwertbildungen auch nur höchst ungenügend erfasste Verhalten realer Quellen legt den Gedanken nahe, das Codierungsverfahren selbst unter Berücksichtigung der Wahrscheinlichkeiten *a posteriori* variieren zu lassen. Erste Überlegungen dazu lassen erkennen, dass so gleichzeitig die praktisch erzielte Widerstandsfähigkeit gegen Kryptanalyse erhöht und die algorithmische Komplexität klein gehalten werden können.

Insbesondere kann man hoffen, so das in Datenschutzanwendungen aktuelle und informationstheoretisch sonst schwer behandelbare Problem in den Griff zu bekommen, dass die zu verschlüsselnden Texte einer insgesamt statistisch inhomogenen, jedoch in Einzelgebieten durchaus sehr homogenen Sprache angehören. Wegen dieses Fehlens von Ergodizität im ganzen sind die Ergebnisse der Shannonschen Überlegungen zur Sicherheit vom Kryptosystemen nur sehr beschränkt auf die im rechnerbezogenen Datenschutz relevante Situation anwendbar.

LITERATUR

1. R. Ash, *Information Theory*, John Wiley und Sons, New York, 1965.
2. K. L. Chung, *Markov Chains with Stationary Transition Probabilities*, Springer-Verlag, Berlin, 1967.
3. F. C. Frick und W. H. Sumby, *Control Tower Language*, J. Acoust. Soc. Amer., Bd. 24, 1952, S. 595-596.
4. F. Jelinek, *Probabilistic Information Theory*, McGraw-Hill Book Co., New York, 1968.
5. F. Jelinek und K. S. Schneider, *Variable-Length Encoding of Fixed-Rate Markov Sources for Fixed-Rate Channels*, I.E.E.E. Trans. Information Theory, Bd. IT-20, 1974, S. 750-755.
6. H. Jürgensen und M. Kunze, *Charakterisierung redundanzfreier Codes zur Datenverschlüsselung*, R.A.I.R.O., Informatique Théorique.
7. T. C. Pasco, *Source Coding Algorithms for Fast Data Compression*, PhD Thesis, Stanford University, CA, 1976.
8. F. Rubin, *Cryptographic Aspects of Data Compression Codes*, Cryptologia, Bd. 3, 1979, S. 202-205.
9. F. Rubin, *Arithmetic Stream Coding Using Fixed Precision Registers*, I.E.E.E. Trans. Information Theory, Bd. IT-25, 1979, S. 672-675.

Zusatz während der Korrektur : Das Dekodierungsproblem wurde von uns mittlerweile in der Arbeit « Complexity of Redundance-Free Codes », Report 142, Dep. Comp. Sci., Univ. Western Ont., 1985, gelöst.