Wolfgang Kastner
Gerhard-Helge Schildt

Informatik
Aufgaben und Lösungen

Dritte, überarbeitete
Auflage

Begleitbuch zu
Blieberger et al.: Informatik

Lehrbuch

SpringerWienNewYork

SpringerWienNewYork

Springers Lehrbücher
der Informatik

Herausgegeben von
o. Univ.-Prof. Dr.-Ing. Gerhard-Helge Schildt
Technische Universität Wien

SpringerWienNewYork

Wolfgang Kastner
Gerhard-Helge Schildt

Informatik
Aufgaben und Lösungen

Dritte, überarbeitete
Auflage

Begleitbuch zu
Blieberger et al.: Informatik

SpringerWienNewYork

Ao. Univ.-Prof. Dr. Wolfgang Kastner
Institut für Rechnergestützte Automation
Arbeitsgruppe Automatisierungssysteme
Technische Universität Wien, Österreich
e-mail: k@auto.tuwien.ac.at

o. Univ.-Prof. Dr.-Ing. Dipl.-Ing. und Ing. (grad.) Gerhard-Helge Schildt
Institut für Rechnergestützte Automation
Arbeitsgruppe Automatisierungssysteme
Technische Universität Wien, Österreich
e-mail: schi@auto.tuwien.ac.at

Printed in Austria

SpringerWienNewYork ist ein Unternehmen
von Springer Science + Business Media
springer.at

Satz: Reproduktionsfertige Vorlage der Autoren
Druck und Bindung: Grasl Druck & Neue Medien, 2540 Bad Vöslau, Österreich

Gedruckt auf säurefreiem, chlorfrei gebleichtem Papier – TCF
SPIN: 10990773

Mit 5 Abbildungen

Bibliografische Information Der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

ISSN 0938-9504
ISBN 3-211-21136-5 SpringerWienNewYork
ISBN 3-211-82861-3 2. Aufl. SpringerWienNewYork

Vorwort zur ersten Auflage

Als die erste Auflage des Buches „Informatik“ vor zwei Jahren erschien, waren Unsicherheiten hinsichtlich der Akzeptanz dieses Buches vorhanden. Mittlerweile ist die erste Auflage vergriffen, und die zweite Auflage erscheint gleichzeitig mit diesem Band. Einer der Kritikpunkte an der ersten Auflage war das Fehlen von Übungsaufgaben am Ende jedes Kapitels. Erst durch die selbständige Erarbeitung von konkreten Beispielen wird es dem Leser möglich, ein tiefes Verständnis der behandelten Materie zu erlangen und sein Verständnis zu überprüfen. Die Aufnahme von Übungsaufgaben in einem sinnvollen Ausmaß würde jedoch den bereits hohen Umfang des Buches weiter erhöhen. Aus diesem Grund entstand der vorliegende Band, der als Ergänzung des Buches „Informatik“ zu verstehen ist und der sich im Aufbau und Inhalt streng nach diesem richtet. Die Aufgaben in diesem Band haben sich im Lehr- und Prüfungsbetrieb seit dem Wintersemester 1989/90 an der Technischen Universität Wien bewährt und spiegeln auch die Schwerpunkte, die im Lehrbetrieb gesetzt wurden, wider. An zahlreichen Stellen wurde Wert auf Erklärungen gelegt, welche über die Beschreibungen im Buch „Informatik“ hinausgehen und somit eine sinnvolle Ergänzung derselben darstellen.

Bester Dank gebührt in erster Linie Maryam Maftoon Kebriai. Ohne ihre Hilfe bei der Erfassung der Beispiele hätte dieses Buch nicht rechtzeitig fertiggestellt werden können. Dank gebührt auch meinen Kollegen Johann Blieberger, Ulrich Schmid und Stefan Stöckler für die großzügige Überlassung zahlreicher Beispiele, sowie Andreas Dluhy für ausgezeichnete Korrekturarbeiten und zahlreichen Studenten für Korrekturen und Anregungen. Nicht zuletzt möchte ich auch Prof. Gerhard H. Schildt für seine wertvollen Hinweise und für seine Unterstützung dieses Vorhabens danken. Trotz Beteiligung zahlreicher Personen an den Erfassungs- und Korrekturarbeiten trage ich die Verantwortung für sämtliche Fehler in diesem Buch. Ich bin für Hinweise auf Fehler und für sonstige Verbesserungsvorschläge dankbar.

Wien, im August 1992 Atilla Bezirgan

Vorwort zur zweiten Auflage

Das Lehrbuch „Informatik“ aus der Lehrbuchreihe der *Informatik* von den Verfassern *Blieberger, Klasek, Redlein* und *Schildt* ist inzwischen in der dritten Auflage erschienen und hat sich am Markt sehr gut eingeführt. In der dritten Auflage war es notwendig, die Inhalte des Informatikbuches an neue informatische Entwicklungen anzupassen: So zum Beispiel die Fuzzy-Logik, wie sie von Lotfi A. Zadeh begründet wurde. Weiter haben sich auch neue Entwicklungen auf dem Gebiet der Betriebssysteme ergeben (so zum Beispiel unter anderem die *Threads* und das damit zusammenhängende *Thread-Scheduling*), aber auch so interessante Fachgebiete wie z.B. die informatische Behandlung des elektronischen Geldes oder Fragestellungen der *Authentisierung*. So stehen derzeit *informationsreduzierende Codierungen* im Vordergrund. Dies war der Anlaß, im Zusammenhang mit dem Vorlesungsbuch Aufgaben und zugehörige Lösungen erneut zu präsentieren.

Der vorliegende Band ist in einen Aufgaben- und Lösungsteil gegliedert. Allen angeführten Antworten liegt der Inhalt des begleitenden Lehrbuches „Informatik“ zugrunde. Es sei darauf hingewiesen, dass im allgemeinen die angeführten Lösungen nicht als absolut angesehen werden sollten, sondern als Musterlösungen gedacht sind. Vor allem bei Rechenaufgaben gibt es oft andere gleichwertige Lösungswege und Notationen.

Zwischenzeitlich haben wir unter Leitung von Herrn Dr. Blieberger die Abwicklung der erforderlichen Übungstests rechnergestützt automatisiert. Dabei werden die Prüfungen bereits vollautomatisch abgewickelt mit Tests, die vom Studierenden direkt auf einem Computer in vorgegebener Zeit zu bearbeiten sind. Darüber hinaus erfolgt auch eine automatisierte Bewertung der Prüfungsleistung durch den Computer. Zusätzlich finden die Studierenden auf der Homepage des *Instituts für Rechnergestützte Automation* unter *www.auto.tuwien.ac.at* Beispiele für übliche Aufgaben des Übungstests. Ziel des vorliegenden Buches mit Aufgaben und Lösungen ist es, diese Basis zu verbreitern und dem Studierenden genügend Anregungen zu geben, um das erlernte Wissen selbständig zu überprüfen.

Unser besonderer Dank gilt Herrn A. Bezirgan, der die Erstellung der ersten Auflage übernommen hat, sowie für redaktionelle Beiträge durch Frau R. Fochtner. Herr Dr. A. Bezirgan hat sich zwischenzeitlich anderen Aufgaben in der Industrie zugewandt; daher war es jetzt notwendig geworden, die zweite Auflage dieses Begleitbuches mit Aufgaben und Lösungen zu erstellen.

Pressbaum, im Oktober 1999 — Gerhard H. Schildt

Vorwort zur dritten Auflage

Das vorliegende Buch „Informatik – Aufgaben und Lösungen“ stellt eine – wie wir meinen – sinnvolle Vertiefung zu dem Lehrbuch „Informatik – Grundlagen“ dar, das inzwischen in der 4. Auflage erschienen ist. Die in das Lehrbuch neu aufgenommenen Kapitel haben uns veranlasst, auch zu diesen Abschnitten entsprechende Aufgaben und Lösungen zur Verfügung zu stellen und die bereits bestehende Aufgabensammlung zu überarbeiten und von Grund auf neu mit weiteren Beispielen zu ergänzen.

Abweichend vom bisherigen Konzept, nur Aufgaben und Lösungen anzubieten, wird in dieser Ausgabe vor jedem Abschnitt das Wesentliche des jeweiligen Themengebietes kurz gefasst präsentiert, um so den Studierenden in die Lage zu versetzen, mit diesen unterstützenden Informationen anschließend die Aufgaben zu bearbeiten und erfolgreich Lösungen zu ermitteln. Dadurch ist dieses Buch in sich geschlossen und hat insofern auch einen eigenständigen Lehrbuchcharakter. Teilweise gehen diese grundlegenden Informationen sogar über den Inhalt des zugrunde liegenden Buches „Informatik – Grundlagen“ hinaus; das liegt jedoch darin begründet, dass es in naher Zukunft wiederum eine Neuauflage geben wird, worin dann auch weiterführende Informationen aufgenommen werden.

Wenn man ein Buch schreibt (und das noch unter enormem Zeitdruck), so verhält es sich ähnlich wie mit dem Prozess der Softwareentwicklung. Allgemein gilt der Satz: *„Software will never be error free!“*. Daher ist den Verfassern bewusst, dass auch einzelne Fehler in der Darstellung enthalten sein können. Wir bitten daher den geneigten Leser, in diesem Fall solche erkannten Fehler an die Adresse `feedback_gdiue@auto.tuwien.ac.at` mitzuteilen. Wir werden bemüht sein, solche Korrekturen in die folgende Auflage aufzunehmen.

Mödling, im Oktober 2004 — Wolfgang Kastner
Pressbaum, im Oktober 2004 — Gerhard Helge Schildt

Inhaltsverzeichnis

1 Informationstheorie

Nach dem Mathematiker Norbert Wiener (1894–1964), dem Begründer der Kybernetik [22], hat Information eine ähnlich grundlegende Bedeutung wie Materie und Energie. Wesentlich für eine Information ist, dass bestimmte Zeichen von einem *Sender* zu einem *Empfänger* gelangen und von diesem dekodiert werden. Die Voraussetzung dafür ist ein gemeinsamer Zeichenvorrat (Alphabet). Bei der Informationsübertragung kann der (materielle) Träger bzw. (energetische) Überträger ohne weiteres gewechselt werden. Beispielsweise kann dieselbe Information einmal geschrieben und einmal gesprochen sein. Unter dem Begriff *Nachricht* versteht man eine Folge von Zeichen, die zur Übermittlung von Informationen dient.

Die von Claude E. Shannon (1916–2001) entwickelte Informationstheorie [19, 20] versucht, ein Maß für den Informationsgehalt zu bestimmen, wieviel Information eine diskrete Nachricht enthält, die von einem Sender an einen Empfänger übermittelt wird. Die Nachricht besteht, wie eingangs erwähnt, aus einer Zeichenfolge, in der die Zeichen mit bestimmten Wahrscheinlichkeiten auftreten.

Die zwei Eigenschaften, die eine Funktion zur Berechnung des Informationsgehalts nach der Shannon'schen Informationstheorie erfüllen soll, sind:

1. Der Informationsgehalt soll nur von der Wahrscheinlichkeit abhängen, mit der ein Zeichen gesendet wird, und nicht von der Art der Codierung. Häufig gesendete Zeichen sollen einen niedrigen Informationsgehalt haben, selten gesendete einen hohen.

2. Der Informationsgehalt einer aus mehreren (voneinander unabhängigen) Zeichen bestehenden Nachricht soll gleich der Summe der Informationsgehalte der einzelnen Zeichen sein.

Der *Informationsgehalt eines Zeichens* ist definiert als der Logarithmus des Reziprokwertes der Wahrscheinlichkeit, mit der das Zeichen auftritt. Als Basis des Logarithmus wählt man 2, wenn es sich um zweiwertige Zeichen handelt. Der Informationsgehalt eines Zeichens wird üblicherweise mit h bezeichnet, die Auftrittswahrscheinlichkeit eines Zeichens mit p. Dann ist

$$h = \mathrm{ld}\,\frac{1}{p} = -\,\mathrm{ld}\,p \tag{1.1}$$

Die Einheit des Informationsgehalts ist *Bit*.

Wenn eine Nachrichtenquelle Zeichen mit unterschiedlicher Wahrscheinlichkeit sendet, so kann der Informationsgehalt eines einzelnen empfangenen Zeichens unter der Voraussetzung berechnet werden, dass man die Auftrittswahrscheinlichkeit des Zeichens kennt. Der Erwartungswert für ein Zeichen ist gleich dem Mittelwert der Informationsgehalte aller Zeichen des zugrunde liegenden Alphabets. Sei $h_i(p)$ der Informationsgehalt des i-ten Zeichens und p die Wahrscheinlichkeit, mit der dieses Zeichen auftritt, so ist der *mittlere Informationsgehalt* H gegeben durch

$$\begin{aligned} H &= \sum_i p_i \cdot h_i \\ &= \sum_i p_i \cdot \frac{1}{\mathrm{ld}\,p_i} \\ &= -\sum_i p_i \cdot \mathrm{ld}\,p_i \end{aligned} \tag{1.2}$$

Dieser mittlere Informationsgehalt wird auch als *Entropie* bezeichnet und in Bit gemessen.

Unter der *mittleren Codewortlänge* L eines Codes versteht man die mit den Auftrittswahrscheinlichkeiten gewichtete Summe der Längen der Codewörter als

$$L = \sum_i p_i \cdot l_i, \tag{1.3}$$

wobei l_i für die Länge des dem i-ten Zeichen entsprechenden Codewortes steht. Im Gegensatz zum Informationsgehalt, der nur durch die Auftrittswahrscheinlichkeiten der einzelnen Zeichen bestimmt ist, hängt die mittlere Wortlänge von der gewählten Codierung ab.

Die mittlere Wortlänge eines Binärcodes ist immer größer oder gleich groß als der mittlere Informationsgehalt eines Codes. Die Differenz zwischen mittlerer Codewortlänge und mittlerem Informationsgehalt (Entropie) ist die *Redundanz* des Codes R.

$$R = L - H, \text{ mit } R \geq 0 \tag{1.4}$$

Die Redundanz eines Codes wird ebenfalls in Bit gemessen.

Die *relative Redundanz* r ergibt sich als bezogene Größe, indem man die absolute Redundanz R auf die mittlere Wortlänge L bezieht als

$$r = \frac{R}{L} \tag{1.5}$$

Für die Berechnung der vorgenannten Größen benötigen wir den *Logarithmus dualis*, der jedoch direkt auf einem Taschenrechner in dieser Form nicht verfügbar ist. Hierzu machen wir folgende Überlegung: Wir gehen von einer Zweierpotenz der Form $a = 2^b$ aus und logarithmieren diesen Ausdruck einmal zur Basis 2 und einmal zur Basis e.

$$\begin{aligned} \operatorname{ld} a &= \operatorname{ld}(2^b) &= b \cdot \operatorname{ld} 2 \\ \ln a &= \ln(2^b) &= b \cdot \ln 2 \end{aligned}$$

Für $b \neq 0$ ergibt sich der Quotient

$$\frac{\operatorname{ld} a}{\ln a} = \frac{\operatorname{ld} 2}{\ln 2}$$

Wegen $\operatorname{ld} 2 = 1$ folgert man

$$\operatorname{ld} a = \frac{\ln a}{\ln 2} \approx \frac{\ln a}{0.693}$$

In einem symmetrischen Binärkanal treten die digitalen Signale '0' und '1' auf. Nimmt man für die Auftrittswahrscheinlichkeit des Zeichens '1' den Wert p an, so beträgt die Auftrittswahrscheinlichkeit des Zeichens '0' den Wert $(1-p)$. Unter dieser Voraussetzung kann man die Entropie H berechnen, die in diesem Zusammenhang als *Shannon'sche Funktion für den symmetrischen Binärkanal* bekannt geworden ist.

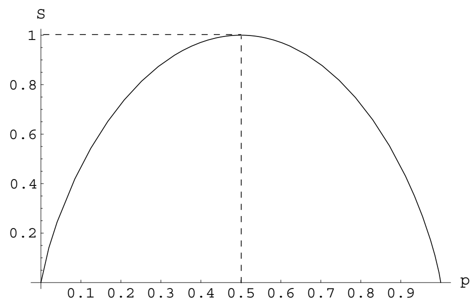
$$H = S = p \cdot \operatorname{ld} \frac{1}{p} + (1-p) \cdot \operatorname{ld} \frac{1}{1-p} \tag{1.6}$$

Es lässt sich zeigen, dass die Entropie H (in diesem Fall S) für gleichwahrscheinlich auftretende Zeichen maximal wird.

Aufgabe 1.1 *Zeichnen Sie den Graphen der* Shannon'schen Funktion $S = f(p)$*!*

Lösung.

p	$p \cdot \mathrm{ld}\,\frac{1}{p}$	$(1-p) \cdot \mathrm{ld}\,\frac{1}{1-p}$	S
0.0	mit Regel von de l'Hospital		0
0.1	0.332193	0.136803	0.468996
0.2	0.464386	0.257542	0.721928
0.3	0.52109	0.360201	0.881291
0.4	0.528771	0.442179	0.970951
0.5	0.5	0.5	1.0
0.6	0.442179	0.528771	0.970951
0.7	0.360201	0.52109	0.881291
0.8	0.257542	0.464386	0.468996
0.9	0.136803	0.332193	0.721928
1.0	mit Regel von de l'Hospital		0



Man erkennt anhand des Funktionsverlaufes, dass die Entropie H im symmetrischen Binärkanal bei $p = 0.5$ maximal wird.

Aufgabe 1.2 *Zeigen Sie, dass die Entropie* H *für gleichwahrscheinlich auftretende Zeichen maximal wird!*

Lösung. Der Nachweis kann dadurch geführt werden, dass man den Extremwert der Shannon'schen Funktion bestimmt, indem man den Differentialquotienten $\frac{dS}{dp}$ bildet, diesen Null setzt, nach dem Extremum sucht und mit Hilfe der 2. Ableitung zeigt, dass es sich um ein Maximum handelt.

$$
\begin{aligned}
f(p) &= p \cdot \mathrm{ld}\,\frac{1}{p} + (1-p) \cdot \mathrm{ld}\,\frac{1}{(1-p)} \\
&= -p \cdot \mathrm{ld}\,p + (p-1) \cdot \mathrm{ld}(1-p) \\
f(p)' &= -\mathrm{ld}\,p - p \cdot \frac{1}{p \cdot \ln 2} + \mathrm{ld}(1-p) + (p-1) \cdot \frac{1}{(p-1) \cdot \ln 2} \\
&= -\mathrm{ld}\,p - \frac{1}{\ln 2} + \mathrm{ld}(1-p) + \frac{1}{\ln 2} \\
&= \mathrm{ld}\,\frac{1}{p} - \mathrm{ld}\,\frac{1}{(1-p)} \\
f(p)'' &= -\frac{1}{p} - \frac{1}{1-p}
\end{aligned}
$$

Um den Extremwert zu finden, wird die erste Ableitung Null gesetzt.

$$
\begin{aligned}
f(p)' = 0 \;\Rightarrow\; -\mathrm{ld}\,p + \mathrm{ld}(1-p) &= 0 \\
\mathrm{ld}\,\tfrac{1-p}{p} &= 0 \\
\tfrac{1-p}{p} &= 1 \\
\Rightarrow \qquad p &= \tfrac{1}{2}
\end{aligned}
$$

Mit Hilfe der 2. Ableitung kann nun herausgefunden werden, ob es sich um ein Maximum handelt.

$$
f(\frac{1}{2})'' = -4 < 0 \Rightarrow \text{Maximum}
$$

Wie auch aus obiger Kurvendiskussion ersichtlich, gilt $H(\frac{1}{2}) = 1$ Bit.

Aufgabe 1.3 *Was ist ein* Bit*?*

Lösung. *Bit* ist die Abkürzung für *binary digit*. Es ist die Einheit des Informationsgehalts und bezeichnet ein Zeichen eines Binäralphabets.

Aufgabe 1.4 *Was versteht man unter dem* Informationsfluss*? In welcher Einheit wird er gemessen?*

Lösung. Der *Informationsfluss* ist die Menge der in einer bestimmten Zeit übertragenen, Information. Seine Einheit ist Bit/s.

Aufgabe 1.5 *Wenn Sie in einem Auto während einer Fahrt jede halbe Stunde auf die Öldruckwarnlampe sehen, wie groß ist dann der Informationsfluss von der Lampe zu Ihnen? Hinweis: Überlegen Sie, wieviel Information Sie bei jedem Hinsehen bekommen.*

Lösung. Die Lampe kann zwei Zustände annehmen: *leuchtend* oder *erloschen*. Somit erhält man bei jedem Hinsehen ein Bit an Information. Man sieht jede halbe Stunde hin, also alle 1800 Sekunden. Der Informationsfluss ist also 1 Bit / 1800 Sekunden = 0.00055556 Bit/s. Beachten Sie, dass wir hier von der Annahme ausgehen, dass der mittlere Informationsgehalt einer Nachricht, die wir von der Lampe bei jedem Hinsehen erhalten, 1 Bit ist. Dies stimmt jedoch bei genauer Betrachtung der Auftrittswahrscheinlichkeiten der einzelnen Nachrichtenzeichen nicht (siehe nächste Aufgabe).

Aufgabe 1.6 *Wenn Sie sich wie im letzten Beispiel verhalten, wie groß ist dann der Informationsgehalt der* leuchtenden *Öldruckwarnlampe? Wie groß ist der Informationsgehalt der* erloschenen *Lampe? Hinweis: Überlegen Sie sich, wie oft die Lampe im allgemeinen in einem intakten Auto leuchtet.*

Lösung. Im Gegensatz zur Annahme im letzten Beispiel haben leuchtende und erloschene Lampen unterschiedlichen Informationsgehalt. Bei einem gut gepflegten, intakten Auto erwartet man, dass diese Lampe praktisch nie leuchtet, d.h., die Auftrittswahrscheinlichkeit einer leuchtenden Lampe ist etwa 0 und jene der erloschenen Lampe nahezu 1. Somit ist der Informationsgehalt einer leuchtenden Lampe praktisch unendlich groß und jener der erloschenen etwa gleich 0.

Aufgabe 1.7 *Wie groß ist der Informationsgehalt jedes einzelnen Zeichens und der mittlere Informationsgehalt in einer Binärcodierung mit* N *Bits, in der alle möglichen Codewörter verwendet werden und gleich wahrscheinlich auftreten?*

Lösung. Da mit N Bits 2^{N} Zeichen dargestellt werden können, gilt

$$\mathrm{h} = \mathrm{ld}\,\frac{1}{\frac{1}{2^{\mathrm{N}}}} = \mathrm{ld}\,2^{\mathrm{N}} = \mathrm{N\ Bit},\ \mathrm{H} = \sum_{\mathrm{i}=1}^{2^{\mathrm{N}}} \frac{1}{2^{\mathrm{N}}} \cdot \mathrm{N} = \mathrm{N\ Bit}$$

Aufgabe 1.8 *Gegeben sei eine Liste von Substantiven mit den dazugehörigen Artikeln. Wenn Sie davon ausgehen, dass die Endung „ung“ ein Substantiv mit dem Artikel „die“ bedingt und die Liste fehlerfrei ist, wie groß ist dann der Informationsgehalt des Artikels des Wortes „Bedeutung“ in der Liste?*

Lösung. Der Informationsgehalt ist 0, da die Information, die man durch den Artikel in einer Tabelle erhält, vom Wort ableitbar und damit redundant ist.

Aufgabe 1.9 *Die Wahrscheinlichkeit für das Auftreten von „Kopf“ oder „Zahl“ beim Münzenwurf beträgt jeweils $\frac{1}{2}$. Wie groß ist die Wahrscheinlichkeit, dass Sie bei drei Würfen dreimal „Zahl“ erhalten? Wie groß ist der Informationsgehalt einer Nachricht, die den Ausgang von drei Würfen beinhaltet, wenn die einzelnen Würfe einander nicht beeinflussen und der Informationsgehalt einer Nachricht, die den Ausgang eines Wurfs beinhaltet,* I *ist?*

Lösung.

$$\mathrm{P}_{\text{dreimal Zahl}} = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{8}$$

Der Informationsgehalt einer Nachricht, die den Ausgang von drei Würfen beinhaltet, wobei die einzelnen Würfe einander nicht beeinflussen, beträgt $3 \cdot \mathrm{I}$.

Aufgabe 1.10 *Bei der Erfassung von Studenten werden siebenstellige Matrikelnummern verwendet. Wie groß ist die Redundanz und die relative Redundanz dieser Codierung bei 10 000 Studenten?*

Lösung. Die Matrikelnummer besteht aus sieben Dezimalziffern. Für die Darstellung aller Studenten braucht man jedoch nur vier Dezimalziffern (0000 ... 9999). Damit sind drei Dezimalziffern überflüssig, d.h.,

$$\mathrm{R} = 3 \cdot \text{ld } 10 \text{ Bit} \quad \text{und} \quad \mathrm{r} = \frac{3}{7}$$

Der Informationsgehalt einer n-stelligen Dezimalzahl ist $n \cdot \mathrm{ld}\, 10$.

Aufgabe 1.11 *Wieviele Fragen brauchen Sie höchstens, um eine natürliche Zahl* < 100*, die ein Freund zufällig gewählt hat, durch Fragen der Art „ist die Zahl größer oder kleiner als* y*“ zu erraten? Wie wählen Sie dabei* y*?*

Lösung. Man benötigt 7 ($\approx$ ld 100 aufgerundet) Fragen. Die erste Frage lautet: „Ist die Zahl kleiner als 50?“. Danach werden die Fragen so gestellt, dass die zu ihnen passenden Antworten entscheiden, in welcher Hälfte des jeweiligen Bereichs die Zahl liegt.

Aufgabe 1.12 *Erstellen Sie eine Logarithmentabelle der ganzzahligen Zweierlogarithmen im Bereich* $[-4, 4]$.

Lösung.

$x = 2^y$	$\mathrm{ld}\ x = y$
16	4
8	3
4	2
2	1
1	0
0.5	-1
0.25	-2
0.125	-3
0.0625	-4

Aufgabe 1.13 *Gegeben ist das Alphabet {?,!,a,5} mit den Auftrittswahrscheinlichkeiten der einzelnen Zeichen. Berechnen Sie den Informationsgehalt* h *jedes Zeichens und den mittleren Informationsgehalt* H *für das Alphabet!*

	p
?	0.125
!	0.625
a	0.1
5	0.15

Lösung.

	p	h
?	0.125	3
!	0.625	0.678072
a	0.1	3.32193
5	0.15	2.73697

$$H = \sum_{i=0}^{3} p_i \cdot h_i \approx 1.5415 \text{ Bit}$$

Aufgabe 1.14 *Wie Aufgabe 1.13 allerdings mit dem Alphabet {x,!,b,7} und folgenden Auftrittswahrscheinlichkeiten:*

	p
x	0.220
!	0.425
b	0.130
7	0.225

Lösung.

	p	h
x	0.220	2.184424571
!	0.425	1.234465254
b	0.130	2.943416472
7	0.225	2.152003093

$$H = \sum_{i=0}^{3} p_i \cdot h_i \approx 1.872065976 \text{ Bit}$$

Aufgabe 1.15 *Berechnen Sie den Informationsgehalt einer (neuen) Nachrichtenquelle nach der Übertragung des Wortes „Sommersemester“!*

Lösung.

Zeichen	p	h	$p \cdot h$
S	3/14	2.22	0.476
O	1/14	3.81	0.272
M	3/14	2.22	0.476
E	4/14	1.81	0.516
R	2/14	2.81	0.401
T	1/14	3.81	0.272

Der Informationsgehalt des Wortes beträgt $H \approx 2.413$ Bit.

Aufgabe 1.16 *Gegeben ist der folgende Binärcode* für das Alphabet $\{A,X,B,Y,G\}$ mit den jeweiligen Auftrittswahrscheinlichkeiten der einzelnen Zeichen.*

	p	Code
A	0.125	1001001
X	0.0625	1011001
B	0.25	1011101
Y	0.0625	1001101
G	0.5	0

Berechnen Sie den Informationsgehalt h für jedes Zeichen, sowie (a) den mittleren Informationsgehalt H, (b) die mittlere Wortlänge L, (c) die Redundanz R und (d) die relative Redundanz r.

Lösung.

	p	Code	h	$p \cdot h$	l	$p \cdot l$
A	0.125	1001001	3	0.375	7	0.875
X	0.0625	1011001	4	0.25	7	0.4375
B	0.25	1011101	2	0.5	7	1.75
Y	0.0625	1001101	4	0.25	7	0.4375
G	0.5	0	1	0.5	1	0.5

(a) $H = \sum_i p_i \cdot h_i = 1.875$ Bit

(b) $L = \sum_i p_i \cdot l_i = 4$ Bit

(c) $R = L - H = 2.125$ Bit

(d) $r = \frac{R}{L} = 0.53125$

Aufgabe 1.17 *Wie Aufgabe 1.16 allerdings mit folgendem Binärcode und Auftrittswahrscheinlichkeiten:*

	p	Code
A	0.25	1000001
B	0.25	1000010
C	0.125	1000011
D	0.125	1000100
E	0.25	0

Schätzen Sie weiter die Redundanz R, begründen Sie Ihre Schätzung und geben Sie eine obere und untere Schranke für R an.

Lösung.

	p	Code	h	$p \cdot h$	l	$p \cdot l$
A	0.25	1000001	2	0.5	7	1.75
B	0.25	1000010	2	0.5	7	1.75
C	0.125	1000011	3	0.375	7	0.875
D	0.125	1000100	3	0.375	7	0.875
E	0.25	0	2	0.5	1	0.25

*Von einem Binärcode spricht man, wenn ein binäres Zielalphabet $\{0,1\}$ vorliegt.

(a) $H = \sum_i p_i \cdot h_i = 2.25$ Bit

(b) $L = \sum_i p_i \cdot l_i = 5.5$ Bit

(c) $R = L - H = 3.25$ Bit

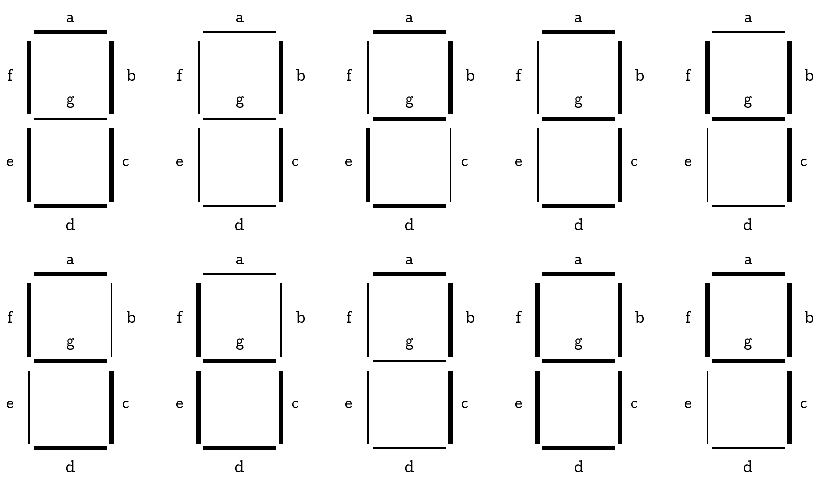
(d) $r = \frac{R}{L} \approx 0.5909$

Eine ganz grobe Schätzung für die Redundanz erhält man durch die Beobachtung, dass die Codewörter für die ersten vier Zeichen des Codes ('A' bis 'D') in den ersten vier Stellen übereinstimmen (`1000xxx`). Da die Summe der Auftrittswahrscheinlichkeiten der angesprochenen Zeichen 0.75 ist, sind in den meisten Fällen mindestens vier Bit überflüssig. Durch die redundanzarme Codierung des Zeichens 'E' wird die Gesamtredundanz gesenkt. Insgesamt ist also die Redundanz in der Nähe von 4.

Durch andere Überlegungen erhält man obere und untere Schranken für die Redundanz. Die ersten vier Zeichen könnte man mit 2 Bit codieren. 5 Bit sind also überflüssig. Da die Codierung von 'E' diese Redundanz nur senken kann, ist 5 Bit eine obere Schranke für die Redundanz. Weiter kann man im Idealfall 5 Zeichen mit ld 5 Bit ≈ 2.32 Bit codieren. Aufgerundet sind das 3 Bit. Die mittlere Wortlänge ist leicht zu berechnen ($L = 0.75 \cdot 7 + 0.25 \cdot 1$) und beträgt 5.5. Daraus folgt, dass (5.5 - 3.0) Bit = 2.5 Bit eine untere Schranke für die Redundanz ist. Die Redundanz liegt also zwischen 2.5 und 5 Bit.

Aufgabe 1.18 *Wie groß ist die* Redundanz R *und die* relative Redundanz r *bei Verwendung einer Siebensegment-Anzeige zur Darstellung der Ziffern 0 bis 9. Gehen Sie davon aus, dass die Ziffern mit gleicher Wahrscheinlichkeit auftreten!*

Lösung. Da jede Ziffer x mit gleicher Wahrscheinlichkeit auftritt, gilt $p(x) = 0.1$. Damit besitzen die Ziffern 0 bis 9 den gleichen Informationsgehalt $h = \text{ld}\,\frac{1}{0.1} \approx 3.32$ Bit. Der mittlere Informationsgehalt ist $H = 3.32$ Bit. Zur Darstellung der Ziffern 0 bis 9 werden sieben Segmente ('a'-'g') verwendet, wobei jedes Segment den Zustand „an" oder „aus" annehmen kann.



Der Entropie steht daher eine mittlere Wortlänge von $L = 7$ Bit gegenüber. Die absolute Redundanz R dieser Codierung beträgt $R = 7 - 3.32 = 3.68$ Bit. Die relative Redundanz ist somit $r = \frac{R}{L} = \frac{3.68}{7} \approx 0.5257$.

2 Codierungstheorie

Unter *Codierung* wird eine Abbildung von einer Sprache auf eine andere verstanden. Einzelne Zeichen oder Zeichenketten der *Quellsprache* werden dabei einzelnen Zeichen oder auch Zeichenketten der *Zielsprache* zugeordnet. Die Abbildung kann, muss aber nicht, umkehrbar eindeutig sein. Bei der Übertragung und Speicherung von Datenmengen ist es sinnvoll, diese Daten in möglichst optimierter Form zu halten. Dadurch können kosten-intensive Ressourcen wie Speicherplatz und Übertragungsbandbreite reduziert werden. Ein Nachteil der Codeoptimierung ist allerdings, dass für das Codieren und Decodieren von Zeichen ein erhöhter Rechenaufwand nötig ist. Daher ist immer genau zu prüfen, welches Codierungsverfahren gewählt wird und ob eine Codeoptimierung überhaupt sinnvoll ist.

Im folgenden Kapitel gehen wir auf die Codierung nach Huffman und Shannon-Fano näher ein, stellen im Anschluss das Verfahren der arithmetischen Codierung vor und diskutieren letztlich fehlererkennende und fehlerkorrigierende Codes.

2.1 Huffman-Code

Ein Verfahren zur Konstruktion eines redundanzarmen Codes wurde von David A. Huffman (1925–1999) entwickelt [10]. Die nach ihm benannte Huffman-Codierung läuft in drei Schritten ab.

1. Die unterschiedlichen Zeichen werden nach ihren Wahrscheinlichkeiten sortiert.*

2. Zu Beginn werden die beiden Zeichen mit den geringsten Auftrittswahrscheinlichkeiten zusammengefasst. Seien diese a und b. Ab diesem Zeitpunkt werden a und b während der Codeerzeugung als *ein* Zeichen mit einer Auftrittswahrscheinlichkeit, die gleich der Summe der beiden Einzelwahrscheinlichkeiten ist, behandelt. Zur Unterscheidung der beiden Zeichen wird für a '0' und für b '1' gewählt (die Zuordnung ist willkürlich). Dadurch entsteht ein binärer Baum der Höhe 1, wobei ein Ast mit '0' und der andere mit '1' gekennzeichnet wird. Die Blätter des Baumes sind mit a und b und seine Wurzel mit dem neuen, durch die Zusammenfassung von a und b entstandenen Zeichen (z.B. ab) beschriftet.

3. Schritt 2 wird solange durchgeführt, bis durch die Zusammenfassung nur noch ein Zeichen übrig bleibt. Durch diese Vorgangsweise entsteht ein binärer Baum. Alle Blätter dieses Baumes sind mit Zeichen des Alphabets belegt. Die Äste des Baumes sind mit '0' und '1' beschriftet. Zum Codewort für ein Zeichen gelangt man, indem die Abfolge der Beschriftungen der Äste des Baumes notiert wird, wobei man von der Wurzel bis zum jeweilig gewünschten Blatt navigiert.

Ein *adaptiver* Huffman-Code [5] ist eine Variation, bei der der Codebaum nach einer gewissen Anzahl von übertragenen Zeichen sowohl beim Sender als auch beim Empfänger, unter Berücksichtigung der relativen Häufigkeit der zuvor tatsächlich übertragenen Zeichen, neu erstellt wird.

*Zeichen mit gleicher Wahrscheinlichkeiten können beispielsweise lexikographisch geordnet werden.

Aufgabe 2.1 *Die Abbildung von Studentendaten wie Namen und Adressen auf die Matrikelnummern des jeweiligen Studenten bildet eine umkehrbar eindeutige Codierung mit Codes fester Wortlänge. Geben Sie je eine andere denkbare Codierung von Studentendaten mit den folgenden Eigenschaften an:*

(a) umkehrbar eindeutig und feste Wortlänge des Codes

(b) nicht umkehrbar eindeutig und feste Wortlänge des Codes

(c) umkehrbar eindeutig und variable Wortlänge des Codes

(d) nicht umkehrbar eindeutig und variable Wortlänge des Codes

Lösung. Mögliche Codierungen von Studentendaten mit den erwünschten Eigenschaften sind:

(a) Solch eine Codierung kann man erhalten, wenn man alle Studenten, die jemals immatrikuliert haben, durchnummeriert, und zwar unter Verwendung einer festen Anzahl von Ziffern. 000000001 würde dann den ersten Studenten kennzeichnen.

(b) Eine Codierung dieser Art erhält man, wenn man die Nachnamen von Studenten als Codewörter verwendet und diese auf eine feste Anzahl von Stellen mit Leerzeichen auffüllt oder abschneidet.

(c) Wie (a), jedoch ohne feste Wortlänge, also 1 für den ersten Studenten und 100 für den hundertsten Studenten.

(d) Wie (b), jedoch ohne Auffüllen bzw. Abschneiden.

Aufgabe 2.2 *Geben Sie bei den folgenden Codierungen jeweils an, ob sie umkehrbar eindeutig sind und die zugehörigen Codes feste Wortlänge haben:*

(a) Morse-Code

(b) LVA-Daten → LVA-Nr. (LVA steht für Lehrveranstaltung)

(c) Telefonbuch (Person → Telefonnummer)

(d) $\mathrm{a} \to 0, \mathrm{b} \to 10, \mathrm{c} \to 01$

Lösung. Die gefragten Eigenschaften der Codierungen sind in der folgenden Tabelle aufgelistet. Die letzte Codierung ist im weiteren Sinne nicht umkehrbar eindeutig, da bei längeren Nachrichten Mehrdeutigkeiten auftreten. Beispielsweise weiß man bei der Nachricht 0101010 nicht, ob sie aus *abbb* oder *ccca* entstanden ist. Eine eindeutige Codierung längerer Nachrichten ist also nicht möglich.

Codierung	umkehrbar eindeutig	feste Wortlänge
Morse-Code	ja	nein
LVA-Daten → LVA-Nr.	ja†	ja
Telefonbuch	nein	nein
$\mathrm{a} \to 0, \mathrm{b} \to 10, \mathrm{c} \to 01$	nein	nein

†Zu jedem Zeitpunkt ist die Zuordnung von LVA-Daten zu LVA-Nr. eindeutig. Es kann sich jedoch über mehrere Semester der Vortragende einer Lehrveranstaltung ändern, ohne dass sich ihre Nummer ändert.

Aufgabe 2.3 *Was versteht man unter einem* Binärcode*?*

Lösung. Ein *Binärcode* liegt genau dann vor, wenn das Zielalphabet der Codierung ein binäres Alphabet ist.

Aufgabe 2.4 *Wenn eine Binärcodierung umkehrbar eindeutig sein soll und wenn die Wortlänge des Codes eine feste Zahl* N *ist, wie groß darf dann das Quellalphabet maximal sein?*

Lösung. 2^{N}, da jede Position in den Codewörtern eines von zwei Zeichen des binären Zielalphabets enthalten kann und somit 2^{N} verschiedene Codewörter existieren.

Aufgabe 2.5 *Wenn umgekehrt das Quellalphabet* N *Zeichen hat und Sie dieses mit einer umkehrbar eindeutigen Binärcodierung mit fester Wortlänge* M *des zugehörigen Codes codieren wollen, wie groß muss dann* M *mindestens sein?*

Lösung. $\mathsf{M}_{\min} = \mathrm{ld}\ \mathsf{N}$, wobei noch aufgerundet werden muss, da es nur ganzzahlige Anzahlen von Binärentscheidungen gibt.

Aufgabe 2.6 *Handelt es sich beim Morse-Code konzeptuell um einen* Huffman-Code*? Begründen Sie Ihre Antwort!*

Lösung. Diese Frage ist nicht eindeutig mit *ja* oder *nein* zu beantworten. Beide Codes sind einander in konzeptueller Hinsicht ähnlich. Bei beiden Codes treten die Zeichen nur an den Blättern auf (beachten Sie, dass das Pausezeichen beim Morse-Code als Zeichenterminator dient). Beide sind Codierungen mit variabler Länge, und beide Codes versuchen, häufig auftretende Zeichen durch kürzere Codewörter darzustellen. Auf der anderen Seite handelt es sich beim Morse-Code um einen ternären Code (die Mächtigkeit des Zielalphabets ist drei). Insgesamt kann man festhalten, dass der Morse-Code große konzeptuelle Ähnlichkeiten zu einem Huffman-Code hat.

Aufgabe 2.7 *Handelt es sich bei den folgenden Codes um* Huffman-Codes*? Begründen Sie Ihre Antwort.*

(a)

	Code
a	*0*
b	*10*
c	*101*

(b)

	Code	p
a	*11*	*0.5*
b	*10*	*0.25*
c	*0*	*0.25*

Lösung. Die angegebenen Codes sind keine Huffman-Codes, da bei diesen Codes

(a) nicht alle Zeichen als Blätter des Codebaumes auftreten (es sind auch nicht alle möglichen Blätter des Baumes besetzt) und

(b) nicht die Zeichen mit kleinsten Auftrittswahrscheinlichkeiten zusammengefasst wurden.

Aufgabe 2.8 *Kann man beim Huffman-Code die Redundanz beliebig klein machen? Wenn ja, wie? Wenn nein, warum nicht?*

Lösung. Ja. Durch die gemeinsame Codierung ganzer Zeichenfolgen.