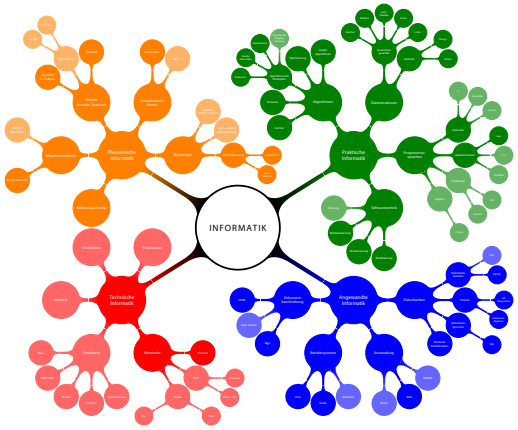
UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR
THEORETISCHE INFORMATIK

Kapitel 1

Informationen und Daten

Wie kommt das Video auf die Scheibe?

Vorlesung Einführung in die Informatik 1 vom 24. Oktober 2013 von Till Tantau



Lernziele von Kapitel 1

1. Die binäre Speicherform verschiedener Daten kennen und verstehen.
2. Wichtige Einheiten von Information kennen und die Speichergröße von Daten abschätzen können.
3. Beliebige Daten kodieren können.
4. Das Konzept der Datenkompression kennen und verstehen.

Gliederung von Kapitel 1

Das Bit ist die kleinste mögliche Informationseinheit.

Computer arbeiten *digital*, das heißt, mit nur zwei Zuständen:

0	versus	1
Strom an	versus	Strom aus
Loch vorhanden	versus	Loch nicht vorhanden
Magnetisierung hoch	versus	Magnetisierung runter

Die Informationseinheit hierzu heißt *Bit* (*B*inary dig*it*).

Mit Bitfolgen lassen sich mehr als nur zwei Möglichkeiten repräsentieren.

Da man mit einem einzelnen Bit nicht viel speichern kann, benutzt man mehrere.

Beispiel (Die vier Nukleotide)

Die vier Nukleotide können mit zwei Bits gespeichert werden:

00	$\hat{=}$	Adenin
01	$\hat{=}$	Cytosin
10	$\hat{=}$	Guanin
11	$\hat{=}$	Thymin

Mit 2 Bits kann man also 4 Dinge unterscheiden.
Mit 3 Bits kann man 8 Dinge unterscheiden.
Mit 8 Bits kann man 256 Dinge unterscheiden.

5-Minuten-Aufgabe

Zur Übung

Lösen Sie eine der folgenden Aufgaben.
(Sie sind nach Schwierigkeit sortiert.)

1. Wie viele Bits braucht man zur Speicherung eines Codons?
2. Wie viele Dinge kann man mit Bitstrings der Länge *genau* n unterscheiden?
3. Wie viele Dinge kann man mit Bitstrings der Länge *höchstens* n unterscheiden?

Bitfolgen bestimmter Längen erhalten besondere Namen.

Bitfolgen bestimmter Längen kommen oft vor und haben deshalb besondere Namen.

Definition (Byte, Kilobyte)

Ein Bitstring der Länge 8 heißt *Byte*.
Ein Bitstring der Länge 8192 heißt *Kilobyte* (kB).

Ein Kilobyte sind also $1024 = 2^{10}$ Byte.

Achtung

Korrekter ist es zu definieren, dass 1000 Byte eine Kilobyte bilden. Deshalb definiert man manchmal genau dies und spricht von »Kibibyte«, wenn man 1024 Byte meint (Kibi = Kilo + Binary).

5-Minuten-Aufgabe

Zur Übung

Schreiben Sie möglichst viele Ihnen bekannte Bezeichnungen für Bitfolgen bestimmter Längen auf.

Wie kann man natürliche Zahlen kodieren?

Problemstellung

Natürliche Zahlen sollen als Bitfolgen dargestellt werden.

Zur Diskussion

Welche Möglichkeiten gibt es? (Es gibt viele!)

Wie kann man ganze Zahlen kodieren?

Problemstellung

Nun sollen auch ganze Zahlen (auch negative) als Bitfolgen dargestellt werden.

Lösungsmöglichkeit

Füge am Anfang der Zahlrepräsentation ein zusätzliches Bit an, das das Vorzeichen angibt.
Beispiel: Aus $6 \mathrel{\hat{=}} 110$ werden $+6 \mathrel{\hat{=}} 0110$ und $-6 \mathrel{\hat{=}} 1110$.

Nachteil: Es gibt zwei Arten, die Null darzustellen ($+0$ und -0). Dies führt unweigerlich zu Programmfehlern.

Für weitere Details, insbesondere zur Kodierung reeller Zahlen, siehe die Literatur.

Wie kann man Texte kodieren?

Beispiel

Wie kodiert man AGGCCCTAAAGT?

Ein zweistufiges Verfahren:

1. Man kodiert die vier möglichen Basen mit zwei Bits:
   00 $\hat{=}$ A
   01 $\hat{=}$ C
   10 $\hat{=}$ G
   11 $\hat{=}$ T
2. Man kodiert Folgen von Basen als Folgen von Bitpaaren:
   Aus AAC wird 000001.
   Aus 010111 wird wieder CCT.

Wie kann man Texte kodieren?

Beispiel

Wie kodiert man "Sehr geehrte Damen und Herren"?

1. Man kodiert die möglichen Buchstaben mit mehreren Bits:
   $000000 \mathrel{\hat{=}} A$
   $000001 \mathrel{\hat{=}} B$
   $000010 \mathrel{\hat{=}} C$
   $\ldots$
   $011000 \mathrel{\hat{=}} a$
   $011001 \mathrel{\hat{=}} b$
2. Wieder kodiert man Texte als Folgen von Bitstrings:
   Aus AAC wird 0000000000000010.

Wie viele Bits sollte man pro Zeichen benutzen?

Lösungsmöglichkeiten:

ASCII Man benutzt 7 Bits, was 128 möglichen Buchstaben entspricht.
Damit kann man die Klein- und Großbuchstaben kodieren, Satzzeichen, und einige Sonderzeichen.

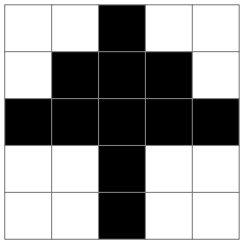
ASCII erw. Man benutzt 8 Bits, womit man 128 zusätzliche Zeichen bekommt.
Diese werden je nach Land anders interpretiert.

UNICODE Man benutzt 16 Bits, womit man 65536 Zeichen kodieren kann.
Dies reicht für alle von Menschen derzeit benutzten Zeichen (inklusive Chinesisch).

ISO 10646 Man benutzt etwa 21 Bits, was für alle jemals benutzte und jemals in der Zukunft benutzte Zeichen ausreicht.

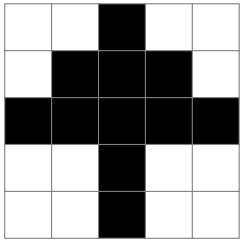
Wie kodiert man Bilder als Bitfolge?

Wie könnte man folgendes Bild kodieren?



Wie kodiert man Bilder als Bitfolge?

Wie könnte man folgendes Bild kodieren?



Mögliche Lösung

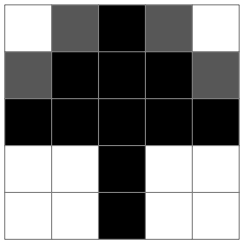
0	0	1	0	0
0	1	1	1	0
1	1	1	1	1
0	0	1	0	0
0	0	1	0	0

Kodierung: 0010000111011111 0010000100
(Zeilen hintereinander weg).

Besser: Stelle Breite und Höhe in den ersten vier Bytes voran.

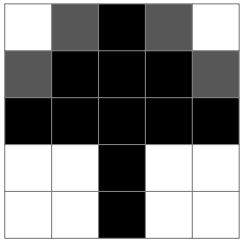
Wie kodiert man Bilder als Bitfolgen?

Wie könnte man folgendes Bild kodieren?



Wie kodiert man Bilder als Bitfolgen?

Wie könnte man folgendes Bild kodieren?

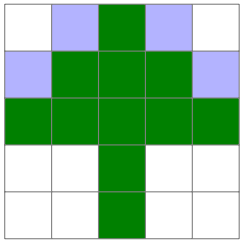


Mögliche Lösung

00	10	11	10	00
10	11	11	11	10
11	11	11	11	11
00	00	11	00	00
00	00	11	00	00

Kodierung: 00101110001011111101111. . .
(Zwei Bit pro Pixel, Zeilen hintereinander weg).

Wie kodiert man Bilder als Bitfolgen?



Idee: Investiere ein Byte für den Rot-Anteil eines Bildpunktes, ein Byte für den Blau-Anteil und ein Byte für den Grün-Anteil.

Beispiel

Rot = (255, 0, 0).
Blau = (0, 0, 255).
Gelb = (0, 255, 255).

Wie kodiert man einen Film?

Problemstellung

Ein Film soll als Bitfolge kodiert werden.

Lösungsmöglichkeit

Ein Film besteht aus 25 Bildern pro Sekunde. Die Kodierung des Films ist dann die Kodierung der einzelnen Bilder hintereinander weg.

Ignorierte Probleme:

- Ton fehlt.
- Untertitel fehlen.
- Nicht fehlertolerant (Kratzer auf der DVD).
- . . .

Eine Beispielrechnung, die ein Problem aufzeigt.

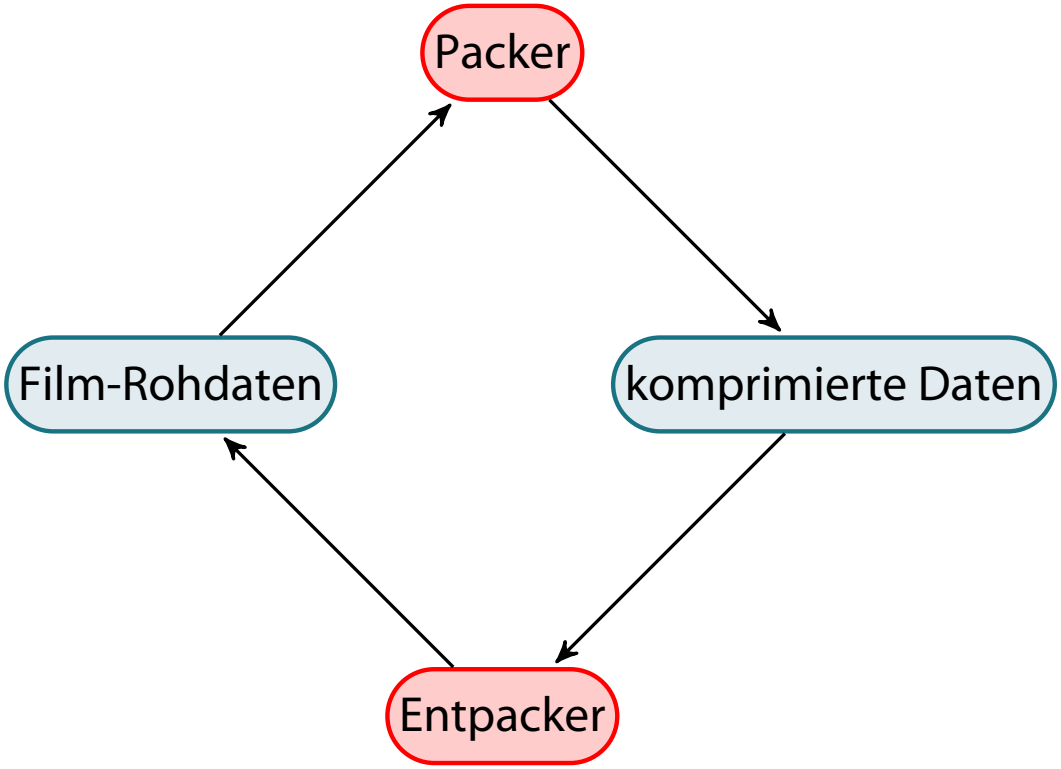
Nehmen wir an, wir wollen einen Spielfilm von 120 Minuten kodieren.

- Das sind 7200 Sekunden.
- Es gibt 25 Bilder pro Sekunde.
- Ein DVD-Bild hat 720 mal 480 Punkte.
- Ein Bildpunkt braucht 3 Byte oder 24 Bit.

Dies macht insgesamt
$7200 \cdot 25 \cdot 720 \cdot 480 \cdot 3 \text{ Byte} = 186624000000 \text{ Byte} \approx 186 \text{ GB}.$

Eine DVD fasst 5 GB bis 16 GB, eine Bluray bis 50 GB.

Lösung des Problems: Datenkompression!



Es gibt zwei Arten von Packverfahren.

Verlustfreie Verfahren

Verlustfreie Verfahren packen Daten so, dass man aus den komprimierten Daten die Originaldaten exakt rekonstruieren kann.
Beispiel: `gzip`, `bzip2`, `zip`

Verlustbehaftete Verfahren

Verlustbehaftete Verfahren packen Daten so, dass man beim Auspacken nur ungefähr die Originaldaten bekommt.
Beispiel: `DivX`, `jpeg`, `mpeg`

Alle Daten sind Bitfolgen

Alle Daten, egal ob Zahlen, Texte, Bilder, Filme oder Musik werden als Bitfolgen gespeichert.

Zusammenfassung

Wichtige Einheiten von Information

Einheit	*Definition*	*Was man damit speichern kann*
Bit	Binary Digit / 0 oder 1	nicht viel
Byte	8 Bits	Buchstaben in unserem Alphabet
Kilobyte (kB)	1024 Byte oder 1000 Byte	kleines Icon, halbe Seite Text
Megabyte (MB)	1024 kB oder 1000 kB	ein Buch, eine Minute Musik, 1 Sekunde HD-Film
Gigabyte (GB)	1024 MB oder 1000 MB	ein Spielfilm, ein Genom
Terabyte (TB)	1024 GB oder 1000 GB	Wikipedia
Kibibyte	immer 1024 Byte	

Datenkompression

- *Verlustfreie* Kompression kodiert (packt) Daten geschickt, so dass die Originaldaten *exakt* rekonstruiert (entpackt) werden können.
- *Verlustbehaftete* Kompression kodiert (packt) Daten geschickt, so dass sie nach dem Entpacken *genauso aussehen oder sich genauso anhören* wie das Original.