





[www.uni-frankfurt.de](http://www.uni-frankfurt.de)

## Modul: Programmierung B-PRG Grundlagen der Programmierung 1 WS 2017/2018

**V 21 Daten – Informationen - Wissen**

Prof. Dr. Detlef Krömker  
Professur für Graphische Datenverarbeitung  
Institut für Informatik  
Fachbereich Informatik und Mathematik (12)

Hier wird Wissen Wirklichkeit



**PRG 1 - Übungen** **EPR - Übungen**

# Herausforderungen, Probleme?

Bitte denken Sie daran, kommenden Montag ihr  
Smartphone oder Laptop mit in die Vorlesung zu  
bringen. Danke.

2 Programmieren 1 – V 06 Prof. Dr. Detlef Krömker

## Unser heutiges Lernziel

*Einige zentrale Grundbegriffe der Informatik kennen lernen:*

**Daten, Information, Wissen** und diverse Details

*Grundlegend ist der Begriff der **Abstraktion** und die damit einhergehende **Modellbildung**.*

**Notation einer Syntax (Struktur einer Information)**

*[Digitalisierung: Abtastung und Quantisierung.]*

*Elemente der **Shannonschen Informationstheorie***

## Übersicht

- Daten und Information
- Digitale Daten – Bits und Bytes
- Information
  - Die syntaktische Ebene
  - Die semantische Ebene
  - Die pragmatische Ebene
- Grundzüge der Shannonschen Informationstheorie
- Syntaxnotationen
- Wissen
- Zusammenfassung
- Ausblick

## Daten und Information

Im mathematisch/formalen Sinn können wie diese Begriffe noch **nicht definieren**, hierzu fehlen uns zu viele Grundlagen

Aber, im informatischen Sinn können wir sie **präzisieren**:

- ihre Verwendung klären,
- ihre Eigenschaften und
- ihre Zusammenhänge kennen lernen.

## Daten und Information

- **Daten** (engl. *data*) ist ursprünglich der Plural von Datum (lat. Singular datum, Plural data = **Gegebenes**)
- (maschinen-) lesbare und bearbeitbare digitale **Repräsentationen von Informationen**, bestehend aus Bits  $\in \{0,1\}$

$$D \xrightarrow{r} I$$

D: Daten

I: Information

$\alpha$ : Zuordnung: Interpretationsvorschrift  
(muss keine Funktion sein  
kann Relation sein)

$$01011000 \left\{ \begin{array}{l} \xrightarrow{r_1} \text{X Buchstabe} \\ \xrightarrow{r_2} 88 \text{ Dezimalzahl} \end{array} \right.$$

## Informationen (einfach)

sind eine Folge von Zeichen deren Elemente einer bestimmten Menge entnommen sind, dem **Zeichenvorrat** (falls geordnet: **Alphabet**).

- **Syntax** (Regelwerk zur Wort oder Satzbildung)
- **Semantik** (Bedeutung)
- **Pragmatik** (Anwendungskontext)

I = „ANTON“ (ein Wort über dem Alphabet der lateinischen Großbuchstaben)

I = 42 (ein Wort über dem Alphabet der Ziffern {0,...,9} in Dezimalschreibweise mit der Semantik ganze Zahl 42)

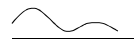
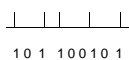
## In der Nachrichtentechnik

Anstelle von **Daten** wird in der Regel der Begriff **Nachricht** benutzt

Das Verhältnis zur Information ist genau dasselbe:

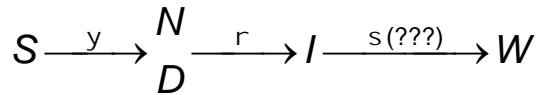
Zur elektrischen oder optischen Übertragung/Speicherung/Berechnung müssen Daten oder Nachrichten als **Signal** repräsentiert sein: **digital** oder auch **analog**

z.B. als Impulsfolge



$$S \xrightarrow{y} N \xrightarrow{r} I$$

## Wissen



also:

nutzbare/anwendbare Signale sind **Daten/Nachrichten**

nutzbare/anwendbare Daten/Nachrichten sind **Informationen**

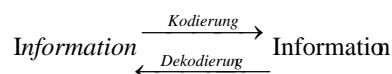
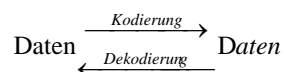
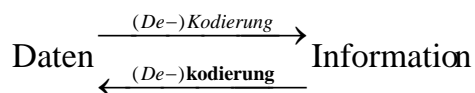
nutzbare/anwendbare Informationen sind **Wissen**, aber ist das eine einfache Relation???

wir müssen prinzipiell „nur“ die Interpretationsvorschriften kennen, um aus Information Wissen zu machen

## Das Begriffspaar: Kodierung – Dekodierung (auch Codierung)

Kodierer/Dekodierer (im Jargon **Codecs** = **CoderDecoder**)

Vorgang, die Interpretationsvorschrift anzuwenden



### Ziele:

- Speicherung
- Übertragung
- Berechnung
- Komprimierung
- Verschlüsselung
- Veranschaulichung

Die einer Codierung zugrunde liegende Abbildung muss **berechenbar, eindeutig und (in der Regel) umkehrbar** sein.

## Beispiele

- |                                 |  |
|---------------------------------|--|
| ▸ bis morgen / see you tomorrow | <b>zwei</b> Nachrichten, die gleiche Information |
| ▸ Ta2-c2                        | <b>Fachsprache:</b> Desoxyribonukleinsäure       |
| ▸ Seelöwe gesichtet             | <b>maskierte Geheimnachricht:</b> z.B. Stichwort |
| ▸ Lirpa                         | sehr einfache <b>Verschlüsselung:</b> April      |

## Digitale Daten – Bits

Ein **Bit** lässt zwei mögliche Werte auf eine Frage zu, z.B.

- Ja oder Nein,
- Wahr oder Falsch,
- Links oder Rechts

Oft werden in diesem Zusammenhang die beiden **Werte 0 und 1 benutzt**, früher auch 0 und L.

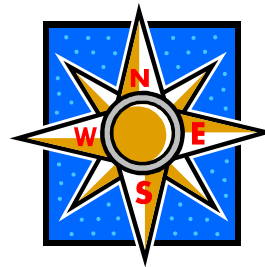
**Bit** ist eine Wortkreuzung aus **binary digit**, englisch für Binärziffer. vorgeschlagen von John W. Tukey (1943) 1946,

In der Fachliteratur 1948 zum ersten Mal von Claude Shannons in seiner berühmten Arbeit „A Mathematical Theory of Communication“ erwähnt.

## Bitfolgen als Code

### Daten (Kode) Information

000 →	Nord
001 →	Nordost
010 →	Ost
011 →	Südost
100 →	Süd
101 →	Südwest
110 →	West
111 →	Nordwest



## Technische Realisierungen

- Die Stellung eines Kippschalters mit zwei Zuständen, zum Beispiel mit den Stellungen EIN oder AUS.
- Der Schaltzustand eines Transistors, "geringer Widerstand" oder "hoher Widerstand".
- Das Vorhandensein einer Spannung, die größer oder kleiner als ein vorgegebener Wert ist, z.B.  $[0..0,8V] = 0$  und  $[2,4..5V] = 1$  (Werte im Bereich  $(0,8...2,4V)$  sind undefinierte fehlerhafte Zustände).
- Magnetisierung (auf einer Festplatte)
- Loch oder Vertiefung in einer CD (Compact Disk)

## Bedeutungen Bit oder bit (Schreibweise uneinheitlich)

- als Element in einer Folge von Binärwerten, meist **Bit**, z.B. Bit 3
- als Stelle in einem nach dem Stellenwertsystem codierten Binärwort meist **Bit** (von z. B. 32 Bit)
- als Speicherzelle, meist **Bit**
- als Einheit für eine Datenmenge, meist **bit**
- als Einheit für den Informationsgehalt (siehe Shannon, später), meist **bit**
- als Datentyp Boole in einigen Programmiersprachen für eine 1 Bit breite Einheit, die True (Wahr) oder False (Falsch) sein kann, dann **Bit**

## Präfixe

..bibit

SI Internationales Einheitensystem ISO/IEC			Binär-Einheiten IEC 60027-2, Second edition, 2000-11		
Name	Symbol	Mehrfaches	Name	Symbol	Mehrfaches
kilobit	kbit	$10^3$	kibibit	Kibit	$2^{10}$
megabit	Mbit	$10^6$	mebibit	Mibit	$2^{20}$
gigabit	Gbit	$10^9$	gibibit	Gibit	$2^{30}$
terabit	Tbit	$10^{12}$	tebibit	Tibit	$2^{40}$
petabit	Pbit	$10^{15}$	pebibit	Pibit	$2^{50}$
exabit	Ebit	$10^{18}$	exbibit	Eibit	$2^{60}$
zettabit	Zbit	$10^{21}$			
yottabit	Ybit	$10^{24}$			



## Anmerkungen

Die angegebenen Präfixe sind die strenge (und korrekte) Version nach IEC 60027-2.

Man muss jedoch feststellen, dass sich die IEC Norm auch im Fachjargon (noch) nicht allgemein durchgesetzt hat.

Also wird man anstelle von gibibit (Gibit) =  $2^{30}$  bit =  $(2^{10})^3$  bit sehr häufig „einfach“ gigabit (Gbit) hören oder lesen.

## Noch eine Einheit: Das Byte

- (fach-)umgangssprachlich die eines Tupels von 8 Bit, deren formale ISO-konforme Bezeichnung **Oktett** wäre. (1 Byte = 8 Bit)
- eine **adressierbare Speichereinheit** in einem Rechensystem
- einen **Datentyp** in einigen Programmiersprachen für eine 8 Bit breite Einheit, die 256 mögliche Zustände annehmen kann
- eine Maßeinheit für 8 Bit bei Größenangaben für **Datenmengen** (Kapazitäten) 1 Byte =  $2^3$  Bit = 8 Bit)
- eine Datenmenge von zwei **Nibbles** (1 Nibble = 4 Bit = 1/2 Byte)
- **gleiche Präfixe wie Bit**

## Byte (Wortherkunft)

Kunstwort vom englisch bit (deutsch: bisschen) und bite (deutsch: Happen).

Speichermenge oder Datenmenge, die ausreicht, um ein alphanumerisches Zeichen (Buchstaben, Zahlen, Sonderzeichen) darzustellen.

1956 von Werner Buchholz geprägt in einer frühen Designphase eines IBM-Computers. (Im Original beschrieb Bite eine Breite von 6 Bit und stellte die kleinste direkt adressierbare Speichereinheit dieses Computers dar)

Bereits Ende 1956 erfolgte aber der Übergang zu 8 Bit. Die Schreibweise Bite wurde zu Byte geändert, um versehentliche Verwechslungen mit Bit zu vermeiden.

Byte wird auch als Abkürzung für „binary term“ erklärt – dabei handelt es sich aber um Backronyme .

Weitere Erklärung: als Kurzform "by eight", frei übersetzt "achtfach".

## Informationen

Es gibt vier Ebenen, unter denen der Begriff „Information“ heute allgemein betrachtet wird. Diese sind

- |                    |   |   |
|--------------------|---|---|
| ▸ <b>Codierung</b> | ↓ | <b>Nachrichtentechnik und Informatik</b>              |
| ▸ <b>Syntax</b>    | ↓ | <b>Formale Sprachen (Th. Informatik) / Linguistik</b> |
| ▸ <b>Semantik</b>  | ↓ | <b>Semiotik und Informatik</b>                        |
| ▸ <b>Pragmatik</b> |   | <b>Kognitionswissenschaft und Informatik</b>          |

## Übersicht

- Daten und Information
- Digitale Daten – Bits und Bytes
- Information
  - Die syntaktische Ebene
  - Die semantische Ebene
  - Die pragmatische Ebene
- **Grundzüge der Shannonschen Informationstheorie**
- Syntaxnotationen
- Wissen
- Zusammenfassung
- Ausblick

## Codierung: $D \xrightarrow{r} I$ oder $N \xrightarrow{r} I$

Beobachtung: Die Zeichenfolge "Informatik ist spannend" ist zu kurz für eine statistische Betrachtung, aber **die Häufigkeit einzelner Buchstaben ist nicht gleich**

Gewisse Buchstaben wie zum Beispiel die Buchstaben *e* und *t* sind häufiger als andere (im Deutschen *e*: 14,70%, *n*: 8,84%, *x*:: 0,01%) (in unserem Beispiel: insbesondere das *n* ist auffallend häufig)

→ das Auftreten von *e* liefert nicht die gleiche „Informationsmenge“ wie *x*

→ bei der Speicherung/Übertragung kann man diesen Sachverhalt zur **verlustfreien Kompression** nutzen, d.h. weniger Daten übertragen, als bei einem adhoc-Code nötig wäre.

## Elemente der Shannonschen Informationstheorie

Die von Shannon formulierte Informationstheorie (1948, A Mathematical Theory of Communication) untersucht **Informationsquellen** mit **statistischen Methoden**.

Er fragte sich, wie man eine **verlustfreie Informationsübertragung**, über elektrische Kanäle (z.B. das Telefonnetz) sicherstellen kann:

- insbesondere, die Datensignale vom Hintergrundrauschen zu trennen und damit **wieviele** Information kann man übertragen
- wie kann man durch einen geeigneten Code **Fehler** bei der Übertragung **erkennen und ggf. korrigieren**

## Beispiel: Codierung von Zeichenfolgen - Buchstabenhäufigkeit

z.B. die Zeichenfolge

"abrakadabra"

- nicht alle Buchstaben kommen gleich häufig vor
- dies gilt generell in allen Sprachen
- im Skript dazu Statistiken

A	absolute Häufigkeit $h_n(A)$	relative Häufigkeit $H_n(A) = \frac{h_n(A)}{n}$
a	5	5/11=0,45
b	2	0,18
d	1	0,09
k	1	0,09
r	2	0,18
Summe (n)	11	1

## Zwischenruf: Algebra der Ereignisse Das Rechnen mit Häufigkeiten und Wahrscheinlichkeiten

Häufigkeit  $\neq$  Wahrscheinlichkeit, aber für große Stichproben kann man die Wahrscheinlichkeit durch Häufigkeiten schätzen ( $\rightarrow$  Statistik, Stochastik)  
Die Rechenregeln sind die gleichen:

Ereignis	Interpretation	Relative Häufigkeit	Wahrscheinlichkeit
$\Omega$	Gesamtereignis (das sichere Ereignis)	$H_n(\Omega) = 1$	$P(\Omega) = 1$
$\emptyset$	Unmögliches Ereignis	$H_n(\emptyset) = 0$	$P(\emptyset) = 0$
A	Beliebiges Ereignis	$0 \leq H_n(A) \leq 1$	$0 \leq P(A) \leq 1$
$A \cup B$	Es treten die Ereignisse A oder B auf	$H_n(A \cup B) = H_n(A) + H_n(B) - H_n(A \cap B)$	$P(A \cup B) = P(A) + P(B) - P(A \cap B)$
und einige weitere, siehe Skript			

## Informationsgehalt einer Nachricht (nach Shannon)

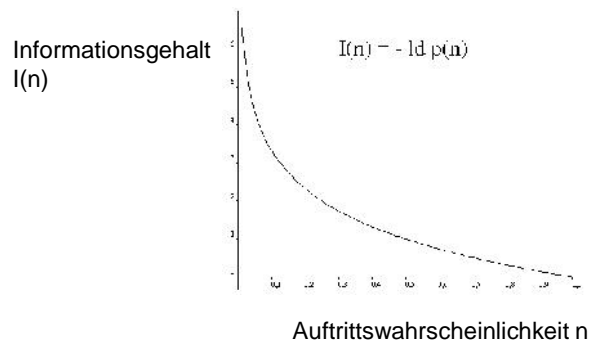
$$I(x) = -\log_2 P(x) = -\lg P(x)$$

negative Logarithmus dualis  
gemessen in bit

### Eigenschaften von $I(x)$ :

- ▶ Mit steigender Wahrscheinlichkeit für das Auftreten einer Nachricht sinkt deren Informationsgehalt.
- ▶ ist immer ein positiver Wert.
- ▶ Für Nachrichten mit gegen Null gehender Wahrscheinlichkeit steigt der Informationsgehalt stark an und geht für  $P(x) \rightarrow 0$  gegen unendlich.
- ▶ Bei unabhängigen Nachrichten addiert sich der Informationsgehalt.
- ▶ **Der Informationsgehalt gibt die günstigste (kürzeste) Codelänge für die Darstellung des Zeichens in Bit an.**

## Informationsgehalt



27

Programmieren 1 – V 06

Prof. Dr. Dettlef Krönker

## Entropie

Shannon definierte die **Entropie  $H$  einer Nachrichtenquelle**  
(Informationsquelle) / über einem Zeichenvorrat  $\Omega$  durch

$$H(I) = - \sum_{j=1}^{|\Omega|} p_j \cdot \log_2 p_j$$

wobei  $p_j$  die Wahrscheinlichkeit ist, mit der das  $j$ -te Zeichen  
des Zeichenvorrats  $\Omega$  im Informationstext / auftritt.

Die **Entropie** erhält die **Einheit bit**.

Maß für die Unsicherheit (des Empfängers)

28

Programmieren 1 – V 06

Prof. Dr. Dettlef Krönker

## Unser Beispiel: „abrakadabra“ sei repräsentativ für das Verhalten der Quelle

Alphabet A	n	absolute Häufigkeit $h_n(A)$	relative Häufigkeit $H_n(A) = \frac{h_n(A)}{n}$	$E = - \sum_{n=1}^{ Q } p_n \cdot \log_2 p_n$ In den untenstehenden Zellen - $p(A) \text{ Id } p(A)$ - $H_n(A) \text{ Id } H_n(A)$
a	1	5	$5/11 = 0,454$	0,517
b	2	2	0,182	0,447
r	3	2	0,182	0,447
k	4	1	0,091	0,314
d	5	1	0,091	0,314
Summe		= 11	= 1	<b>E = 2,040</b>

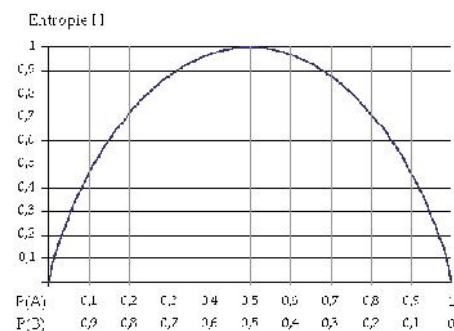
## Interpretation

- Die Entropie dieser Nachrichtenquelle ist 2,04 bit  
= mittlere Information / Zeichen
- Theoretisch ist bei dieser Häufigkeitsverteilung der Zeichen einer Quelle eine Kodierung der 5 verschiedenen Zeichen mit 2,04 Bit pro Zeichen erreichbar, das wäre eine **Optimalcodierung**.
- Das heißt mit anderen Worten: Es gibt keine Codierung dieser Informationsquelle, die mit zwei Bit auskäme, man braucht leider etwas mehr.
- Man kann sich dem Wert 2,04 aber praktisch annähern, indem man verschieden lange Codewörter benutzt: für die häufig auftretenden Zeichen wählt man kurze Codewörter, für die seltener auftretenden Zeichen entsprechend längere Codewörter

## Diskussion der Formel $H(I) = - \sum_{j=1}^{|\Omega|} p_j \cdot \log_2 p_j$

- **Minimalwerte:** Sofern immer nur eine bestimmte Nachricht eintrifft, also die Wahrscheinlichkeit für deren Auftreten gleich 1 ist, erreicht die Entropie einer Nachrichtenquellen ihren Minimalwert. Es ergibt sich ein Betrag von  $(1 \text{ Id } 1) \rightarrow 0$ . Die Nachrichtenquelle stellt keinerlei Information zur Verfügung, es besteht keine Unsicherheit über die vorraussichtlich nächste Nachricht.
- Wenn eine bestimmte Nachricht von der Nachrichtenquelle überhaupt nicht ausgesendet wird, ergibt sich für diese Nachricht ein Betrag von  $(0 \text{ Id } 0) \rightarrow 0$ , d.h. auch sie trägt nicht zur Information auf Seiten des Empfängers bei.
- **Maximalwerte:** Die Entropie einer Nachrichtenquelle erreicht ihren Maximalwert, wenn alle Nachrichten mit der gleichen Wahrscheinlichkeit auftreten. In diesem Fall besteht die größtmögliche Unsicherheit über die Natur der zu erwartenden Nachricht. Folglich ist bei deren Eintreffen der Informationsgewinn am größten.

## Der zugehörige Graph: Entropie einer binären Nachrichtenquelle mit den Zeichen A und B:



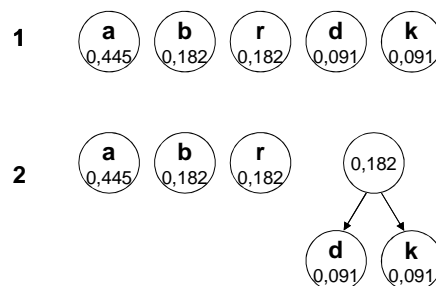
$$H(I) = - \sum_{j=1}^{|Z|} p_j \cdot \log_2 p_j$$



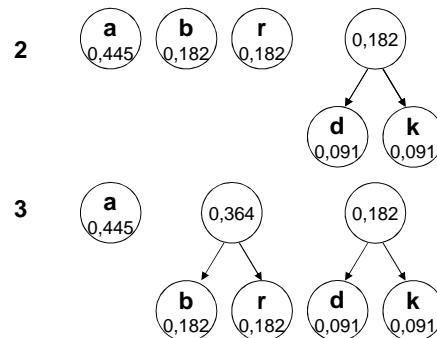
## Anwendung bei der Codeentwicklung → Hamming Code

A	relative Häufigkeit	$-p(A) \lg p(A)$ $= -H_n(A) \lg H_n(A)$	kanonischer Code Blockcode
a	$5/(11 \cdot 100) = 0,45$	0,517	<b>000</b>
b	0,18	0,447	<b>001</b>
r	0,18	0,447	<b>010</b>
k	0,09	0,314	<b>011</b>
d	0,09	0,314	<b>100</b>
Summe	1	<b>2,040</b>	<b>→ 3 bit</b>

## Entwicklung eines Codes mit variabler Codelänge: Huffman Code – 1. und 2. Schritt



### Entwicklung eines Codes mit variabler Codelänge: Huffman Code – 3. Schritt

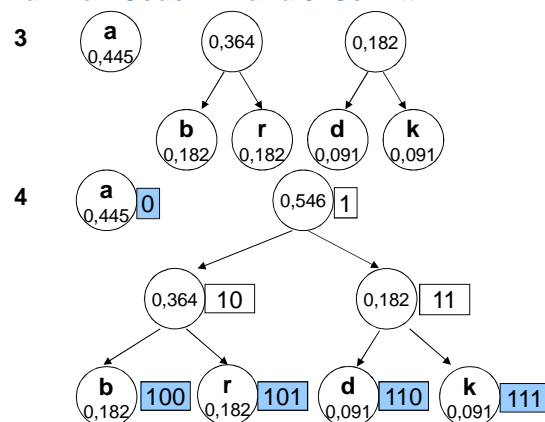


35

Programmieren 1 – V 06

Prof. Dr. Detlef Krönker

### Entwicklung eines Codes mit variabler Codelänge: Huffman Code – 4. und 5. Schritt



36

Programmieren 1 – V 06

Prof. Dr. Detlef Krönker

## Das Ergebnis

A	relative Häufigkeit	$-p(A) \log_2 p(A)$ $= -H_n(A)$	Blockcode	Huffman	Gewichtete Codelänge
a	0,45	0,517	000	<b>0</b>	0,455
b	0,18	0,447	001	<b>100</b>	0,545
d	0,09	0,314	010	<b>110</b>	0,273
k	0,09	0,314	011	<b>111</b>	0,273
r	0,18	0,447	100	<b>101</b>	0,545
Summe	1	<b>2,040</b>	<b>3 bit</b>		<b>2,091 bit</b>

37

Programmieren 1 – V 06

Prof. Dr. Detlef Krönker

## Diskussion

Die Huffman Codierung ist eine Entropiecodierung, ein Optimalcode.

Die **rein statistische Berechnung** der informationstheoretischen Entropie ist **gleichzeitig ihre Beschränkung**.

Beispielsweise ist die Wahrscheinlichkeit, eine 0 oder 1 in einer **regelmäßigen** Zeichenkette "1010101010..." zu finden, genauso groß, wie in einer Zeichenkette, die durch statistisch unabhängige Ereignisse (etwa wiederholten Münzwurf) entstanden ist.

Shannons Entropie für beide Zeichenketten sind identisch, obwohl man intuitiv die erste Kette als weniger zufällig bezeichnen würde  
 → algorithmischen Entropie

38

Programmieren 1 – V 06

Prof. Dr. Detlef Krönker

## Übersicht

- Daten und Information
- Digitale Daten – Bits und Bytes
- Information
  - Die syntaktische Ebene
  - Die semantische Ebene
  - Die pragmatische Ebene
- Grundzüge der Shannonschen Informationstheorie
- **Syntaxnotationen**
- Wissen
- Zusammenfassung
- Ausblick

## Informationen

Es gibt vier Ebenen, unter denen der Begriff „Information“ heute allgemein betrachtet wird. Diese sind

- |                    |   |  |
|--------------------|---|--|
| ▸ <b>Codierung</b> | ↓ | Nachrichtentechnik und Informatik              |
| ▸ <b>Syntax</b>    | ↓ | Formale Sprachen (Th. Informatik) / Linguistik |
| ▸ <b>Semantik</b>  | ↓ | Semiotik und Informatik                        |
| ▸ <b>Pragmatik</b> |   | Kognitionswissenschaft und Informatik          |

## Syntax

**Information** wird **nur als Struktur** gesehen, die es zu übermitteln gilt.

Die Bedeutung ist hierbei im Wesentlichen uninteressant.

Als Beispiel können hier **Regeln** angeführt werden, wie aus Zeichen Worte zu bilden sind (Formale Sprachen) oder aus Worten Sätze (natürliche Sprachen): eine **Grammatik** (Regeln vom Satzbau)

Grundprinzip der syntaktischen Information ist die **Unterscheidbarkeit**:

**Information** enthält, was unterschieden werden kann.

## Beschreibung einer Grammatik durch BNF oder EBNF

Durch eine **formale Grammatik** wird festgelegt, ob es sich bei einer **formalen Sprache** (z. B. **Programmiersprache**) um erlaubte Konstruktionen oder unerlaubte Konstruktionen handelt.

EBNF ist eine erweiterte BNF (Backus-Nauer-Form):

Backus-Nauer Form (erstmals genutzt zur Beschreibung von Algol 60)

Erweiterungen von Nikolaus Wirth (1977)

Heute genormt in **ISO/IEC 14977: (1996E)**, siehe Moodle

## Eine Syntax wird durch eine formale Grammatik beschrieben

durch eine „Metasprache“

Eine formale Grammatik besteht aus

- **Terminalsymbolen (terminals),**
- **Nichtterminalsymbolen (nonterminals) und**
- **Ableitungsregeln (Produktionen)**
- **einem Startsymbol**

## Terminalsymbole

Ein **Terminalsymbol** bzw. **Terminalzeichen** ist ein Symbol, das (einzeln) nicht weiter durch eine Produktion ersetzt werden kann.

Mit anderen Worten, ein Terminalsymbol kommt auf keiner linken Regelseite einzeln vor, ist elementar, atomar (siehe nachfolgend).

In der Theorie (der Informatik) werden Terminalsymbole häufig mit Kleinbuchstaben gekennzeichnet, z.B. „a“ oder „b“

in der Praxis jedoch auch andere Symbole wie Satzzeichen und Schlüsselwörter von Programmiersprachen, z.B. „FOR“ oder „IF“, usw.

Terminalsymbole werden in Anführungszeichen gesetzt: “ “

## Ableitungsregeln

- In einer **Ableitungsregel** (Ersetzungsregel) wird das Zeichen = (oder die Zeichenfolge ::=) zur Definition benutzt, endet immer mit;
- das Zeichen | (vertikaler Strich) dient zur Kennzeichnung von Alternativen;
  - Beispiel: **Alternative:**  
Ziffer außer Null = "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9";
- das Zeichen , (Komma) dient zur Kennzeichnung von Sequenzen;
- Beispiel: **Sequenz:**  
Ziffer = "0" | Ziffer außer Null;  
Zweistellige Zahl = <Ziffer außer Null>, <Ziffer>;
  - Eine Ziffer ist also eine 0 oder eine Ziffer außer Null.  
Eine zweistellige Zahl ist eine Ziffer außer Null gefolgt von einer Ziffer.

## Nichtterminalsymbole

Ein Nichtterminalsymbol wird in BNF durch Ableitungsregeln definiert:

stehen dort auf der linken Seite des =

## Ableitungsregeln in BNF (Defizite)

- Wiederholungen muss man in BNF über eine Rekursion definieren. Eine Ableitungsregel kann dazu auf der rechten Seite das Symbol auf der linken Seite enthalten.

- Beispiel: **Rekursion**  
 $\langle \text{Ziffernfolge} \rangle ::= \langle \text{Ziffer} \rangle \mid \langle \text{Ziffer} \rangle, \langle \text{Ziffernfolge} \rangle$

Also: *Eine Ziffernfolge ist eine Ziffer oder eine Ziffer gefolgt von einer Ziffernfolge.*

## BNF → EBNF

Es hat sich aber gezeigt, dass die BNF teilweise umständliche Konstrukte benötigt, um z.B. optionale Elemente, also Elemente, die ausgelassen werden dürfen, sowie sich wiederholende Elemente darzustellen.

Niklaus Wirth hat in der Definition der Sprache Pascal zusätzliche Elemente eingeführt um diese Schwierigkeiten zu umgehen und nannte dies extended BNF (EBNF, erweiterte BNF); I

In der Literatur sind verschiedene Varianten der Backus-Naur-Form gebräuchlich, wir benutzen im Weiteren

Die für Python genutzte Form ist etwas anders als die genormte Variante siehe Skript und Übung



## Symbole in EBNF (1) nach ISO/IEC 14977 : 1996(E)

Metalanguage symbol	Normal representation
concatenate-symbol	, comma
defining-symbol	= equals sign
definition-separator-symbol	vertical line
start-comment-symbol	(*
end-comment-symbol	*)
start-option-symbol	[
end-option-symbol	]
start-repeat-symbol	{
end-repeat-symbol	}

## Symbole in EBNF (2) nach ISO/IEC 14977 : 1996(E)

Metalanguage symbol	Normal representation
first-quote-symbol	'
second-quote-symbol	"
start-group-symbol	(
end-group-symbol	)
repetition-symbol	*
terminator-symbol	;
special-sequence-symbol	?

## Häufiger Gebrauch;

- ▶ Aber um zum Beispiel die Syntax einer Programmiersprache zu beschreiben, benutzt man oft vereinfachte Formen, z.B. bei Python: **Modified BNF grammar notation**, siehe:
- ▶ <https://docs.python.org/3/reference/introduction.html>, 1.2 Notation
- ▶ Für Codecs, etc. in Networks nutzt man oft

### Abstract Syntax Notation One (ASN.1)

auch ein Internationaler Standard (ISO): letzte Revision: 5.0 Edition (2015)  
kommt aber aus der welt der CCITT (Telekommunikation)

## Python Syntax Notation

<https://docs.python.org/3/reference/introduction.html>, 1.2 Notation

Metalanguage symbol	Representation
defining-symbol	::= two colon and equal
alternative symbol	vertical line least binding
zero or more repetitions (of the preceding item)	* star tight binding
one or more repetitions	+ plus tight binding
zero or one occurrences (the enclosed phrase is optional)	[ ] a phrase in square brackets
grouping	( ) parantheses
informal description	<...> a phrase in angular brackets

white space is only meaningful to separate tokens

## Semantik

### Bedeutungsebene

Dazu muss ein bestimmtes Bezugssystem angelegt werden, um die Strukturen in eine Bedeutung überführen zu können.

Dieses Bezugssystem bezeichnet man auch als Code. Im Beispiel "Informatik ist spannend" muss man also "wissen", was „*spannend*“ bedeutet, um den Sinn zu erfassen.

Interessant aber auch klar:

**Information kann weitergegeben werden, ohne dadurch weniger zu werden.** ... Mit Materie oder Energie geht das nicht.

## Pragmatik

Die Aussage, dass Informatik spannend ist, hat **echten Informationscharakter**, wenn wir uns mittags um 8 Uhr nach einer durchzechten Nacht noch halb schlaftrunken überlegen, ob wir wirklich in die Vorlesung gehen sollen [Freitags ;-)]

Der pragmatische Informationsgehalt der - semantisch exakt gleichen - Aussage ist aber gleich null, wenn wir um 10.30 Uhr bereits im Hörsaal sitzen und einer spannenden Erläuterung lauschen [Freitags ;-)]

„Smalltalk“ ist eine Art des Informationsaustausches (-rauschens), bei dem die offensichtlich über die Sprache ausgetauschten semantischen Informationen so gut wie keine pragmatische Information darstellen.

## Zusammenfassung Information (allgemein)

- Information ermöglicht die **Verringerung von Ungewissheit**.
- **Information ist übertragbar**; in Form von Daten bzw. Nachrichten - hierzu werden Signale genutzt
- Information ist ein Ereignis, dass den **Zustand des Empfängers bzw. Systems verändert**, seine Unsicherheit reduziert
- Information führt zu einem **Gewinn an Wissen**.

## Übersicht

- Daten und Information
- Digitale Daten – Bits und Bytes
- Information
  - Die syntaktische Ebene
  - Die semantische Ebene
  - Die pragmatische Ebene
- Grundzüge der Shannonschen Informationstheorie
- Syntaxnotationen
- **Wissen**
- Zusammenfassung
- Ausblick

## Nur ganz kurz und einfach: Wissen



- Wissen steht in der griechischen Philosophie im **Gegensatz zur Meinung**. Demnach impliziere Wissen Wahrheit und könne durch keine Argumentation widerlegt werden, während eine Meinung zwar wahr sein könne, aber diskutabel sei.
- Wissen** = subjektiv und objektiv zureichendes Fürwahrhalten
- Glauben** = subjektiv zureichendes, objektiv unzureichendes Fürwahrhalten
- Meinen** = subjektiv und objektiv unzureichendes Fürwahrhalten

## Wissen

**drei Aussagen, die alle drei erfüllt sein müssen!**

- Dem Wissen liegen Informationen zugrunde, oder anders formuliert: Wissen ist mit Bedeutung (Semantik) **und** Pragmatik ausgestattete Information.
- Diese Informationen müssen derart aufeinander bezogen sein, dass sie nachvollziehbar in sich stimmig sind, also ohne Widersprüche!
- Neben der inneren Übereinstimmung muss sich Wissen in Übereinstimmung mit den wahrnehmbaren Bedingungen einer Umwelt (der realen Welt, aber auch der sozialen Gemeinschaft: Wissen ist kein individuelles Relikt) befinden.

## Wie kann Wissen im Computer repräsentiert werden?

Wir unterscheiden:

- deklaratives Wissen versus prozedurales Wissen
- diskursives Wissen versus narratives Wissen
- explizites Wissen versus implizites Wissen
- informal – semi-formal – formal
- usw.

## Wie kann Wissen im Computer repräsentiert werden?

zentrales Thema des Informatik Teilgebietes „**Künstliche Intelligenz**“

(KI, *artificial intelligence* oder *AI*).

Viele alternative Repräsentationen, z.B.: (künstliche) Neuronale Netze, Semantische Netze, Frames, Begriffliche Graphen, Beschreibungslogiken, Ontologien / Semantic Web.

**aber auch viele „Pleiten“**

„**starke KI**“ (*strong AI*) und „**schwache KI**“ (*weak AI*)

Die starke KI hat zum Ziel, eine künstliche Intelligenz zu erschaffen, die wie der Mensch **kreativ nachdenken und Probleme lösen kann** und die sich durch eine Form von **Bewusstsein** beziehungsweise Selbstbewusstsein sowie **Emotionen** auszeichnet. Bis heute nicht gelöst?

## Starke KI



## Schwache KI

Im Gegensatz zur starken KI geht es der **schwachen KI** darum, konkrete (Anwendungs-) probleme zu meistern.

Insbesondere sind das solche Probleme, zu deren Lösung nach  
allgemeinem Verständnis irgendeine Form von „Intelligenz“ notwendig  
zu sein scheint.

## Simulation intelligenten Verhaltens mit Mitteln der Informatik

sehr viele gute Lösungen, aber das führt hier zu weit!

## Zusammenfassung

Wir haben einige zentrale Grundbegriffe der Informatik kennen gelernt:

„Daten“ und „Information“ und die zugehörigen Einheiten (Bit/bit)

Kodierung – Syntax – Semantik – Pragmatik

Grundzüge der Shannonschen Informationstheorie

... jetzt müssen Sie nacharbeiten und üben!

## Fragen und (hoffentlich) Antworten



## Ausblick (kommende Woche)

- Selbstprogrammierte Klassen und Typen in Python

**Danke für Ihre Aufmerksamkeit**