

# Der Begriff der REDUNDANZ

Copyright © by V. Miszalok, last update: 06-12-08

- ↓ [Einleitung](#)
- ↓ [Definition von INFORMATION](#)
- ↓ [Beispiele fuer INFORMATION](#)
- ↓ [Definition von REDUNDANZ](#)
- ↓ [Beispiele für REDUNDANZ](#)
- ↓ [Vergleich der REDUNDANZ-Beispiele](#)
- ↓ [Feststellungen](#)
- ↓ [Unterteilung der Informatik](#)
- ↓ [Begriffliche Probleme von INFORMATION und REDUNDANZ](#)
- ↓ [Genetischer Code](#)
- ↓ [Links](#)

## Einleitung

Das Wort REDUNDANZ stammt von dem lateinischen Verb „redundare“, was bedeutet überfließen, im Überfluss vorhanden sein. Redundanz kann entweder positiv Überfluss im Sinne von Reichtum oder aber negativ überflüssiger Ballast bedeuten. Diese Doppelbedeutung macht den Begriff für die Informationstheorie interessant. Wir werden sehen, dass REDUNDANZ gleichzeitig Reichtum und Ballast bedeutet, je nachdem wer der Empfänger ist.

## Definition von INFORMATION

Die folgenden Definitionen sind (stark vereinfacht) entnommen aus dem Beitrag von Warren Weaver in dem klassischen Grundlagenbuch von

C.E. Shannon: "The Mathematical Theory of Communication", Univ. of Illinois Press, 1949.

Definition 1a: INFORMATION ist die Mindestanzahl von Bits, die zur Codierung einer Nachricht verwendet werden müssen.

Definition 1b: INFORMATION ist die Länge des kürzest denkbaren Codes, mit dem man eine Nachricht übertragen kann.

## Beispiele für INFORMATION

Beispiel 1.1: Ein Wetterbericht, der sich beschränkt auf die beiden Mitteilungen "1 = Sonne" und "0 = keine Sonne" hat die INFORMATION von 1 Bit.

Beispiel 1.2: Ein Wetterbericht, der 4 Alternativen mitteilt, "11 = Sonne" "10 = bedeckt" "01 = Regen" "00 = sonstiges" hat die INFORMATION von 2 Bit.

Der informationstheoretische Begriff „INFORMATION“ unterscheidet sich somit grundlegend von dem umgangssprachlichen Begriff „Information“.

Der Begriff INFORMATION hat also nichts mit dem Inhalt zu tun, sondern bezeichnet die Mindestmenge von Nullen und Einsen, die notwendig ist, um einen Sachverhalt aus einer Menge möglicher Sachverhalte mitzuteilen. Wenn die Menge alternativer Sachverhalte groß ist, dann braucht man viele Bits zur Identifikation eines bestimmten Sachverhalts und die INFORMATION der Nachricht ist dann groß.

Beispiel 1.3: Es gibt auf der Welt mehr Telefonanschlüsse als Kreditkarten. Die INFORMATION einer Telefonnummer muss folglich höher sein als die einer Kreditkartennummer (obwohl die Kreditkartennummern normalerweise länger sind als die Telefonnummern und obwohl für Diebe eine Kreditkartennummer interessanter ist als eine Telefonnummer).

## Definition von REDUNDANZ

Ist ein Code länger als unbedingt notwendig (Länge des Codes größer als INFORMATION der Nachricht) dann enthält der Code REDUNDANZ.

**Definition:** REDUNDANZ ist der Logarithmus zur Basis 2 des Quotienten aus Codelänge dividiert durch INFORMATION.

Formel:  $REDUNDANZ = \log_2 ( \text{Codelänge} / \text{INFORMATION} )$ .

Die REDUNDANZ kann niemals negativ werden. Sie ist Null, wenn der Code so kurz wie möglich ist (Codelänge identisch mit INFORMATION, das ist selten so). Sie ist Eins, wenn die Codelänge 2 mal so groß ist wie INFORMATION. Sie ist größer als Eins, wenn der Code viel länger ist als INFORMATION (das ist unter Menschen und allgemein bei der Kommunikation unter Biosystemen der Normalfall).

## Beispiele für REDUNDANZ

**Beispiel 2.1:** Wir verändern unseren Wetterbericht indem wir ein 8-Bit-ASCII-Zeichen übermitteln: **S** = Sonne, **B** = bedeckt, **R** = Regen, **?** = sonstiges.

Die INFORMATION bleibt unverändert 2 Bit, die Codelänge steigt von 2 auf 8 Bit. Die REDUNDANZ dieses Codes beträgt  $\log_2(8/2) = \log_2(4) = 2$ .

**Beispiel 2.2:** Wir verändern unseren Wetterbericht, indem wir in sieben 8-Bit-ASCII-Zeichen übermitteln: **Sonne** = Sonne, **bedeckt** = bedeckt, **Regen** = Regen, **???????** = sonstiges.

Die INFORMATION bleibt unverändert 2 Bit, die Codelänge steigt von 2 auf 7\*8 Bit. Die REDUNDANZ dieses Codes beträgt  $\log_2(56/2) = \log_2(28) = 4,8$ .

**Beispiel 2.3:** Wir verändern unseren Wetterbericht indem wir Icons (=32x32 Pixel Minibildchen) übermitteln: Die INFORMATION bleibt unverändert 2 Bit, die Codelänge steigt von 2 auf 32\*32\*8 Bit. Die REDUNDANZ dieses Codes beträgt  $\log_2(32*32*8/2) = 5+5+3-1 = 12$ .



## Vergleich der REDUNDANZ-Beispiele

Vergleichen wir Beispiel 1.2 mit Beispiel 2.3:

	Beispiel 1.2 mit 2-Bit-Code	Beispiel 2.3 mit Icon-Code
INFORMATION	2 Bit	2 Bit
Codelänge	2 Bit	$8*32*32 = 8192$ Bit
REDUNDANZ	$\log_2(1) = 0$	$\log_2(4096) = 12$
Übertragungszeit, Telefonkosten	minimal	deutlich
Kleiner Übertragungsfehler	zerstört alles	spielt keine Rolle
Eignung für Nachrichten zwischen Menschen	schlecht, Handbuch unbedingt erforderlich	optimal, erklärt sich selbst, keine Sprachkenntnisse notwendig, auch für Analphabeten geeignet
Eignung für Nachrichten zwischen Computern	optimal. Ein Auswerteprogramm braucht nur 3 if-Abfragen.	null, weil es so viele Bildmöglichkeiten für Sonne etc. gibt, dass kein Programm alle erfassen kann.

## Feststellungen

Zur Unterscheidung weniger Zustände braucht man wenig INFORMATION.

Zur Unterscheidung vieler Zustände braucht man viel INFORMATION.

Codiert man INFORMATION, muss man sich für eine REDUNDANZ entscheiden.

REDUNDANZ Null ist fast immer ungünstig, weil redundanzlose Codes extrem störanfällig und unleserlich sind. Je höher die REDUNDANZ umso fehlertoleranter wird der Code und umso besser wird er von Menschen verstanden.

Je höher die REDUNDANZ umso mehr Speicherplatz und Übertragungsbandbreite braucht der Code und umso schwerer wird er von Computern verstanden.

Menschen mögen REDUNDANZ, Computer hassen REDUNDANZ.

Codes mit niedriger REDUNDANZ ( Telefonnummern, Autokennzeichen, Kontonummern etc) sind für Menschen ungeeignet.

Codes mit hoher REDUNDANZ ( Sprache, Bilder, Musik ) sind für Computer ungeeignet.

Computer können die Redundanz aus natürlichen Codes wie Bildern oder Sprache nicht entfernen.

Es ist enorm schwer, automatisch REDUNDANZ zu vernichten und die INFORMATION zu destillieren.

REDUNDANZ "klebt" fest am Code.

Bei allen Eingaben in die Maschine muss der Mensch die ihm natürliche REDUNDANZ weitgehend vernichten (Tastatur).

Bei allen Ausgaben an den Menschen muss die Maschine die ihr natürliche REDUNDANZ-Armut überwinden und redundanzreiche Codes erzeugen (Monitor).

Es gibt dumme Redundanz (= mehrfache identische Wiederholung) und intelligente Redundanz (= kunstvolle Variation).

Musik ist z.B. reine Redundanz. Sie enthält keine Nachricht und kann trotzdem voller Geist und Genie sein.

**Das zentrale Problem der Schnittstelle zwischen Maschine und Mensch ist die Anpassung von REDUNDANZ.**

Die Tatsache, dass jeder Computer seinen Besitzer zwingt, redundanzarm mit ihm zu reden, weckt Bedenken, ob nicht der Besitzer vor seinem Computer zwangsläufig verblödet. Er muss sich kurzfassen, logische Sprünge und Assoziationen unterlassen, alles muss er nackt und einfach formulieren. Das ist unmenschlich. Aber es ist wie beim Fahren und Fliegen. Durch den Mangel an Körperbewegung sterben ja unsere Beine nicht ab. Der massenhafte Müll im Internet ist noch kein Beleg dafür, dass das Netz Gehirne physiologisch geschädigt hat.

Vergleich	Computer	Biosystem
Input [bit/sec]:	1 Giga	100 Giga
Output [bit/sec]:	1 Giga	1 Kilo
Redundanz:	0 bis 4	10 bis 14
Basiscode:	0 und 1	unbekannt
Speicherform des Codes:	magnetisch	unbekannt
Transportform des Codes:	Rechteckimpulse	Membranpotentiale
Operationen mit Zahlen:	gut	<b>schlecht</b>
Operationen mit Bild+Ton:	nur via Zahlen	gut
Fähigkeit zur identischen Wiederholung:	sehr gut	null
Eindeutigkeit:	fast immer	fast nie
Störsicherheit:	null	gut
Lernfähigkeit:	schlecht	gut

Computer kennen keinen Spass an sprunghaften Assoziationen, Offenheit fürs Abgelenktwerden und freuen sich niemals über Redundanz.



Copyright [Thilo Rothacker](#)

## Unterteilung der Informatik

Parallel zu den Rasterdisplays entwickelt sich seit ca. 1980 aus der alten Informatik (= Datenverarbeitung = Data Processing) eine neue Informatik an der Mensch-Maschinen-Schnittstelle. Man unterscheidet an Hand des Begriffes der REDUNDANZ folgende Teilgebiete:

1. **Data Processing** = Datenverarbeitung = klassische Informatik = Umwandlung von Zahlen und Buchstaben in andere Zahlen und Buchstaben = Umwandlung von Codes niedriger REDUNDANZ in andere Codes niedriger REDUNDANZ.

Beispiele: Finanzwirtschaft, Verwaltung, Datenbanken, Computernetze, Mathematik, Statistik

2. **Computer Graphics** = Computer Graphik = Umwandlung von Zahlen in Bilder = Umwandlung von Codes niedriger REDUNDANZ in Codes hoher REDUNDANZ.

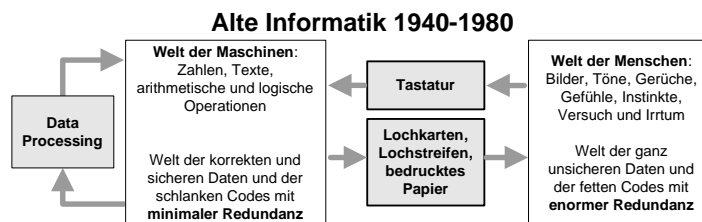
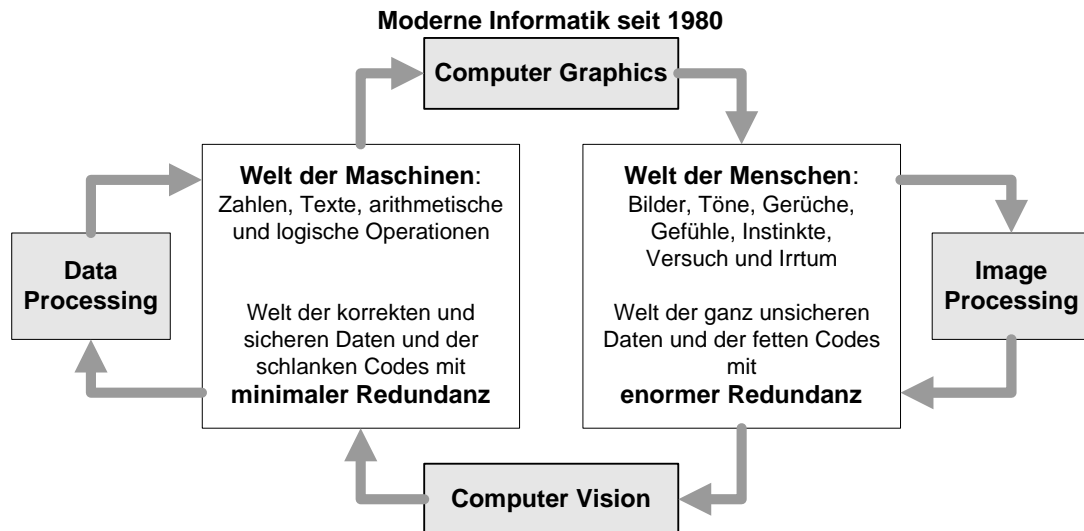
Beispiele: Graphical User Interface = GUI von Mac+Linux+Windows, Technisches Zeichnen = Computer Aided Design = CAD, Ersatz von Zahlen = Business Graphik, Entertainment = Spiele+Trickfilm

3. **Image Processing** = Bildverarbeitung = Umwandlung von Bildern in andere (bessere) Bilder = Umwandlung von Codes hoher REDUNDANZ in andere Codes hoher REDUNDANZ.

Beispiele: Scanner, Kopierer, DigitalTV, CT, Ultraschall, Radar

4. **Computer Vision** = Computer Sehen = Mustererkennung = Pattern Recognition = Umwandlung von Bildern in Zahlen = Umwandlung von Codes hoher REDUNDANZ in Codes niedriger REDUNDANZ = REDUNDANZ-Vernichtung.

Beispiele: BarCodeLeser, Belegleser = Optical Character Recognition = OCR, Erkennen von Autokennzeichen, Fingerabdrücken, Chromosomen, Tumoren, Flugkörpern etc.



## Begriffliche Probleme von INFORMATION und REDUNDANZ

1. Problem: Die beiden Definitionen von INFORMATION sind weit entfernt von dem, was man umgangssprachlich als Information versteht. Die Nachrichten "In der Spree schwimmen Delphine." und "In der Spree schwimmen Plastiktüten." können unmöglich die gleiche Information haben. Aber zu ihrer minimalen Codierung braucht man ungefähr gleich viele Bit. Ausweg: Shannon (siehe oben) schlägt vor, den Kehrwert der Wahrscheinlichkeit (=die Unwahrscheinlichkeit einer Nachricht) als Maß für INFORMATION zu benutzen. Der Vorschlag ist genial aber praktisch schwer umzusetzen. Man müsste alle Alternativen kennen und diese müssten disjunkt (unabhängig voneinander) sein, um die Wahrscheinlichkeit einer speziellen Nachricht angeben zu können.

2. Problem: Die Begriffe INFORMATION und REDUNDANZ sind nur in der Nachrichtentechnik exakt fassbar. Außerhalb der Nachrichtentechnik gelingt es selten, INFORMATION und REDUNDANZ zu quantifizieren. Das bedeutet aber nicht, dass die Begriffe wertlos wären. Auch ohne Quantifizierung sind die beiden Begriffe unersetzlich, wenn man die vielfältigen Umwandlungsprozesse (=Umcodierungen) von Nachrichten zwischen Mensch und Medien begreifen will.

Beispiel: Der gewaltige Roman von Leo Tolstoj "Krieg und Frieden" ist sehr oft verfilmt worden in mancherlei Qualität. Die Übertragung des Romantextes (etwa im Internet) braucht bei weitem weniger Bits als die Übertragung eines der Filme. Das würde bedeuten, dass selbst im schlechtesten Film mehr INFORMATION enthalten wäre als im Roman. Die Wahrheit aber ist, dass die Filme INFORMATION vernichten, weil sie nur zentrale Bruchstücke des Romans darstellen und die Bruchstücke bildlich gewaltig aufblähen. Die Verfilmung ist also Vernichtung von INFORMATION bei gleichzeitiger Erzeugung enormer REDUNDANZ.

3. Problem: Es gibt ein tiefsitzendes Vorurteil, dass Computeraktivität und menschliches Denken ähnliche Vorgänge seien. In Wahrheit sind die Analogien nur oberflächlich und die Unterschiede enorm. Die Biosysteme sind evolutionär optimiert auf die Verarbeitung starker Datenströme, die voller Störungen sind. Die Computer sind evolutionär optimiert auf die Verarbeitung dünner Datenströme, die aber so gut wie keine Fehler enthalten dürfen.

Beispiel: Ein menschliches Auge enthält 120 Mio Sinneszellen (= ca. 50 GigaBit/sec pro Auge), vor denen Blutgefäße und ein fehlerbehaftetes optisches Abbildungssystem sitzen und beobachtet von der schwankenden Plattform des knöchernen Schädels aus eine zwischen Tag und Nacht, Schatten und Sonne wechselnde Umwelt in der sich Nahrung und Gefahr verbergen.

Die Begriffe INFORMATION und REDUNDANZ erhellen die Unterschiede: Computer sind INFORMATIONs-verarbeitend und deshalb hilflos gegen Störungen, Biosysteme sind REDUNDANZ-verarbeitend und deshalb unglaublich störsicher. Die Fähigkeit zur Kooperation derartig unterschiedlicher informationsverarbeitender Systeme muss man realistisch niedrig einschätzen: Menschen können niemals Freunde von Computern werden, die Mensch-Maschinen-Schnittstelle wird immer eine Schmerzquelle bleiben.

**Die Verminderung dieser Schmerzen ist die zentrale Aufgabe der Medieninformatik.**

## Genetischer Code

Das Genom des Menschen ist eine Kette aus 3,2 Mrd. Doppelbits 00, 01, 10, 11. Die Kette hat also 6,4 Mrd. Bits oder 800 MByte und passt auf eine CD. Jede der  $10^{13}$  bis  $10^{14}$  Zellen meines Körpers besitzt diese CD und ein eigenes Schreib-Lese-Laufwerk.

Neben dem Genom des Menschen sind auch die einiger Tier- und Pflanzenarten fast vollständig sequenziert und es stellt sich heraus, dass die CD weder eine Datei-Struktur noch ein Inhaltsverzeichnis besitzt. Der seit Milliarden Jahren von der Evolution geschriebene Code ist zwar beim Menschen auf 46 [Chromosomen](#) verteilt aber sonst ein strukturloses LaLuLa ohne Kapitel, Absatz, Punkt und Komma. Es ist unglaublich schwierig herauszufinden, welche biologische Eigenschaft wo codiert ist.

Das Hauptproblem ist die hohe Redundanz des Genoms, sie beträgt mindestens  $\log_2(128) = 7$  vielleicht sogar  $\log_2(1024) = 10$ .

Fast alle Eigenschaften sind mehrfach (mindestens zweifach) an verschiedenen Stellen codiert. Die Codes an diesen Stellen sind sehr ähnlich aber nicht identisch.

Mehrfach codiert sind wahrscheinlich Abschnitte, an denen Experimente stattfinden. Die Kopien sind zur Mutation freigegeben, während das Original konserviert bleibt.

Man kann aus der Länge eines sinnvollen Abschnittes weder auf die Wichtigkeit noch auf die Komplexität schließen. Außerdem können Gene ihre Positionen wechseln, wandern, und verkehrt und zufällig irgendwo im Genom eingebaut werden. Bei der Eigenreduktion der Chromosomen nach der Zellteilung treten erstaunliche Kopierfehler auf. Wenn Codestücke mehrfach vorkommen, ist dies kein Hinweis auf deren Wichtigkeit. Auch längst vergangener Schutt ist mehrfach codiert.

Hochkomplizierte Steuerungen der Embryonalentwicklung (Hox-Gene) können kurz redundanzarm codiert sein, unwichtige Enzyme dagegen lang und kompliziert.

Innerhalb eines Gens können sich lange Sequenzen identisch wiederholen oder solche Wiederholungen können fehlen ohne dass das Gen Funktion gewinnt oder einbüßt.

Es gibt offensichtlich keinen speziellen Bauplan des Menschen sondern unser Genom enthält fast alle Pläne für fast alles, was je gelebt hat. Den meisten Code teilt der Mensch mit allen Lebewesen, den Tomaten, Hefen, Bakterien und Insekten und mit längst ausgestorbenen Spezies. Wie Microsoft legt die Evolution großen Wert auf Abwärtskompatibilität und verwendet uralten Code in neuen Betriebssystemen weiter. Ca. 80 % des menschlichen Genoms ist offensichtlich identisch mit dem der Fruchtfliege Drosophila, 98,4% sind identisch mit jeder Affenart. Kaum glaublich, dass in 1,6 % des Genoms der göttlichen Funke verborgen sein soll, nämlich die Fähigkeit zur Sprache, zur Musik und zur Mathematik. Die Sonderstellung des Menschen beruht wahrscheinlich eher auf seinem Reichtum an im Datenmüll eingestreuten "Exonen", nämlich Datenschnipsel, die keine Proteine sondern Ribonukleinsäuren = Genschalter codieren.

Wir sind deshalb keine Tomaten oder Drosophilas, weil vor fast jedem Gen ein hochkomplexes Schaltfeld dessen Aktivierung bewacht. Die Schalter werden eingestellt von wandernden Ribonukleinsäuren und erst diese komplizierten Signalmolekül-Kaskaden machen uns zu Menschen. → [Epigenetics](#)

**Es ist nicht zu bestreiten: Die Evolution schreibt weit bessere Betriebssysteme als alle Informatiker in Seattle und Stanford.**

Es liegt nicht an der Codemenge: Windows Vista Professional benötigt nicht nur eine, sondern vier CDs.

Es liegt nicht an der Redundanz: 90% der Funktionen von Vista kennt und benutzt kaum jemand.

Es liegt an der Individualität: Eine Vista- oder Linux-Installation darf keine Individualität besitzen. Sie muss den exakten Clon eines Musters herstellen; sein Zustand ist zementiert. Anders als Microsoft verteilt die Evolution keine exakten Kopien (=Clone) sondern kopiert ziemlich schlampig und erzeugt Varianten.

Rätsel: In der Informatik konnte man noch nie ein schlechtes Programm durch mehrfaches (auch nicht durch Millionen-faches) schlampiges Kopieren verbessern. Niemand weiß, warum das in der Natur (auf lange Sicht) funktioniert.

Die Lokalisierung von Funktionen im Genom und die Aufklärung der Signalketten in deren Schaltfeldern wird die zentrale wissenschaftliche Aufgabe des 21. Jahrhunderts. Die Aufgabe lautet: Entferne die REDUNDANZ aus dem redundanzreichen Code und destilliere die INFORMATION.

Die Aufgabe wird dann gelöst sein, wenn die REDUNDANZ auf Null gesunken sein wird (was wahrscheinlich unerreichbar, aber auch nicht unbedingt notwendig ist). Je trennschärfer wir die REDUNDANZ entfernen, umso klarer werden wir die Zusammenhänge erkennen und umso treffsicherer werden wir unseren eigenen Code schreiben und uns selbst konstruieren.

Wir jetzt Lebenden sind wahrscheinlich die vorletzten Urmenschen. Die Befreiung aus dem Tierreich, aus Krankheit, Dummheit, Alterszerfall und sogar Tod wird sich, so Gott will, innerhalb der nächsten 200 Jahre materiell vollziehen. Tausend Jahre Gesundheit, Jugend und eine Durchschnittsintelligenz jenseits derzeitiger Vorstellungskraft werden real für die Enkel unserer Enkel. Wir wollen hoffen, dass die Zukünftigen diejenigen Urmenschen nicht verachten und vergessen werden, die heute mit beschränktem Verstand mühsam aber fleißig die Grundlagen legen.

## Links

Fragen und Antworten: [www.miszalok.de/Questions/QA01\\_Redundancy/FA\\_Redundanz.htm](http://www.miszalok.de/Questions/QA01_Redundancy/FA_Redundanz.htm)

Download: [Witziges über Störsicherheit im Kopf \(PDF 6 kB\)](#)