

## Beispiel 1: Entropie

Um die mittlere Entropie des strings *abracadabra* zu bestimmen, müssen wir zunächst die Häufigkeit der einzelnen Zeichen ermitteln:

Zeichen	$n_{char}$
a	5
b	2
r	2
c	1
d	1

Table 1: Häufigkeit der Symbole im string

Die Wahrscheinlichkeit, dass dieses Zeichen vorkommt, erhalten wir indem wir die Häufigkeit  $n_{char}$  durch die Anzahl der Symbole im string ( $N = 11$ ) dividieren. Wenn wir diese Funktion in die Formel aus der Vorlesung einsetzen erhalten wir:

$$H = - \sum_{i=0}^5 \frac{n_{char,i}}{N} \log \left( \frac{n_{char,i}}{N} \right) \quad (1)$$

Durch das Auswerten der Summe erhalten wir eine mittlere Entropie  $H$  von 2.04. Wenn wir nun die Gesamt-Entropie bestimmen wollen, müssen wir das Resultat mit der String-Länge  $N$  multiplizieren. Somit hat der string "abracadabra" eine Gesamt-Entropie von 22.4.

## Beispiel 2: State-Machines (Endlicher Automat)

State-Machines können im allgemeinen in zwei Hauptkategorien unterteilt werden:

1. Acceptors (Akzeptoren)
2. Transducer (Transduktoren)

Acceptors werden häufig für Wort-Erkennung verwendet und bestehen im Wesentlichen aus zwei states: accepted und rejected. Dabei werden oft mehrere Kriterien nacheinander getestet, um einen Input zu prüfen.

Transducer generieren einen Output der vom derzeitigen Zustand der State-Machine und gegebenenfalls von einem zusätzlichen Input abhängt. Ein Beispiel hierfür ist eine Ampel, die nach einer vom Zustand abhängigen Zeit auf einen anderen Zustand wechselt.

## Beispiel 3: Programmiersprachen

Wie bereits der Name vermuten lässt, muss eine compilierte Sprache compiliert werden. Das bedeutet, dass der geschriebene Code in Maschinensprache (also in eine Reihe von Befehlen, die der Computer versteht) übersetzt wird und ein ausführbares Programm erzeugt wird. Beispiele: C, C++, Fortran, usw.

Bei interpretierten Sprachen wird das Programm direkt ausgeführt. Dabei wird der geschriebene Code Zeile für Zeile in eine andere Sprache übersetzt und ausgeführt. Beispiele: Python, R, Matlab, usw.

## Beispiel 4: Global Interpreter Lock

Der GIL ist ein Mutex, der bewirkt, dass immer nur ein Thread Code-Zeilen eines python-Programms ausführen kann. Dies erschwert die Implementation von multi-threading, ist aber ein wichtiger Bestandteil von python.

In python besitzen Variablen einen Referenz-Zähler. Sobald dieser auf Null herabgesetzt wird, wird die Variable aus dem Speicher gelöscht. Wenn mehrere Threads Code-Zeilen ausführen, kann dieser Counter simultan erhöht oder erniedrigt werden (race-condition), was dazu führen kann, dass die Variable fälschlich gelöscht wird. Dies führt natürlich zu Problemen beim Ausführen des Programms.

## Beispiel 5: Binär und Hex-System

z.B. 25 in Binär und Hex:

$$11001 = 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 = 25 = 1 * 16^1 + 9 * 16^0 = 19$$