

Multimedia Processing: Übungszettel 1

29.9.2016

1. Basiswechsel (2 Punkte). Schreiben Sie ein Programm, das eine Zahl von einer Basis in eine andere Basis konvertiert. Es ist ausreichend wenn das Programm für Basen von 2 bis 9 funktioniert.
Syntax: `convert <zahl> <basis_von_zahl> <ziel_basis>`
2. Basiswechsel-Erweiterung-, 2 Punkte. Erweitern Sie das Programm zum Basiswechsel, so dass es für beliebige Basen (größer 2) funktioniert. Verwenden Sie einen Beistrich als Trennzeichen für jede Basis größer 10. Z.b. wird die Zahl 64 in der Basis 16, dann als „4,0“ dargestellt, weil $4 * 16^1 + 0 * 16^0 = 64$.
3. Gegeben sei ein Zeichenvorrat $Z = \{A, F, I, M, N, O, R, T\}$ mit folgender Codierung (4 Punkte):

Zeichen	Code
A	01001
F	10
I	01000
M	0111
N	0101
O	11
R	0110
T	00

Dekodieren Sie folgenden Code.

010000101101101100111010010001000110101

Schreiben Sie ein Programm, dass Bitfolgen dieses Codes automatisch dekodiert.

Fehlerhafte Codes sollen erkannt werden.

Syntax: `decode <string>`

4. Relative Häufigkeit, 5 Punkte. Schreiben Sie ein Programm, das die relativen Häufigkeiten der einzelnen Zeichen einer Text-Datei bestimmt. Output format: “<ascii_code>\t<relative_frequency>”
Syntax: `relativ_frequency <file_to_analyse> <result_csv>`
5. Entropie, 2 Punkte. Schreiben Sie ein Programm, das basierend auf den Auftretswahrscheinlichkeiten der einzelnen Zeichen (den relativen Häufigkeiten) die Entropie berechnet.
Syntax: `entropy <result_csv>`