# Einführung in die Sprachwissenschaft

## Übung 1

Oktober 2020

Lösen Sie die folgende Übungen individuell. Wenn die Anzahl der Übereinstimmungen in den gewählten Beispielen größer als durch Zufall zu erwarten ist, müssen Sie die Aufgabe erneut einreichen, oder wird diese als nicht-bestanden bewertet.

Laden Sie die Lösung als eine PDF-Datei in Moodle hoch. Bei 40 bis 45 Punkte bekommen Sie eine 1,0; bei 20 Punkte eine 4,0.

1. Wählen Sie ein willkürliches Wort, dass mindestens ein Zeichen enthält, dass nicht zum ASCII-Zeichensatz gehört. Schreiben Sie für jeden Buchstaben dieses Wortes die Zahl in hexadezimal, dezimal und binär auf. Nutzen Sie hierfür Codepage 850, Codepage 852, Codepage 437 oder ISO 646. Die Tabellen finden Sie z.B. auf Wikipedia. (10 Punkte)

   Ich habe Codepage 437 gewählt.

   Größe

   | Wort: | G | r | ö | ß | e |
   |---|---|---|---|---|---|
   | Binär | 0100 0111 | 0111 0010 | 1001 0100 | 1110 0001 | 110 0101 |
   | Hexadezimal | 47 | 72 | 94 | E1 | 65 |
   | Dezimal | 71 | 114 | 148 | 225 | 101 |

   Leider haben die meisten von Ihnen nicht angegeben, welche Kodierung sie benutzt haben. Es konnten für diese Aufgabe aus 4 Kodierungen gewählt werden. Etwa die Hälfte von Ihnen hat aber ISO-8859-1 gewählt, das nicht zur Auswahl stand.

2. Schreiben Sie für jede Zahl aus der vorigen Aufgabe auf, wie diese Zahl in ISO 8859-1 interpretiert wird. (10 Punkte)

   | Hexadzimal | 47 | 72 | 94 | E1 | 65 |
   |---|---|---|---|---|---|
   | Buchstabe | G | r | ? (nicht definiert) | á | e |

3. Ist die Folge der Binärzahlen aus der ersten Übung eine korrekte Folge in UTF8? Wenn nein, argumentieren Sie, warum das nicht der Fall ist. Wenn ja, schreiben Sie die Buchstaben auf, die dort stehen, wenn man die Zahlen als UTF8 Reihenfolge liest. (5 Punkte)

   Nein: das dritte Byte fängt mit der Folge 10 an und ist demzufolge in UTF8 in Folgebyte. Das Byte davor hätte also mit 11 anfangen müssen. Das ist aber nicht der Fall. Die Folge ist also ungültig.

4. Suchen Sie einen Text (Oder Chatnachricht, oder E-Mail) in dem ein Zeichen falsch dargestellt ist. Fügen Sie den Text (kurzer Ausschnitt mit dem Fehler reicht) oder ein Screenshot in Ihrer Abgabe ein. Rekonstruieren Sie erstens mit welcher Kodierung der Text ursprünglich

geschrieben wurde, und zweitens, wie der Text bei der Darstellung interpretiert wurde. Zeigen Sie hierbei möglichst genau, welche Zahl falsch interpretiert wurde. (10 Punkte)

```
B³rozeiten wõhrend der Ferien:
   * 8:30 Uhr - bis 12:00 Uhr
   * 14:00 Uhr - bis 16:30 Uhr
```

Die falsch dargestellte Zeichen sollten natürlich ü und ä sein. In ISO-8859-1 sind die Codes für ü und ä: FC und E4. In Codepage 850 stehen diese Codes genau für die in der Abbildung dargestellte Zeichen. Der Text wurde also in ISO-8859-1 gespeichert und dann in Codepage 850 dargestellt. (Die Lösung ist relativ einfach zu finden, das nur wenige Codepages das hochgestellte 3 enthalten).

Noch ein Beispiel (eins reicht natürlich):

```
BÃ¼rozeiten wÃ¤hrend der Ferien:
   * 8:30 Uhr - bis 12:00 Uhr
   * 14:00 Uhr - bis 16:30 Uhr
```

Es werden zwei Zeichen an der Stelle eines Buchstabens dargestellt. Die Ursprüngliche Kodierung muss also UTF8 sein, da UTF8 die einzige Kodierung ist (von denen die wir kennen gelernt haben), in der Buchstaben teilweise durch mehr als einem Byte dargestellt werden. Die Buchstaben ü und ä werden in UTF8 kodiert als 11000011 10111100 und 11000011 10100100, oder Hexadezimal: C3 BC und C3 A4. Diese 3 Codes werden in ISO-8859-1 wie in der Abbildung dargestellt. Der Text wurde also in ITF8 gespeichert und dann in ISO-8859-1 dargestellt.

5. Wie viel Byte pro Buchstaben braucht man etwa zum Speichern eines einfachen englischen Textes, wenn man reines Unicode verwenden würde? Und wie viel zum Speichern eines russischen Textes? Wie viel zum Speichern eines philippinischen Textes (Tagalog)? Siehe auch:`http://www.utf8-zeichentabelle.de/unicode-utf8-table.pl` (5 Punkte)

   Englisch: 1 Byte
   Russisch (Kyrilisch): 2 Byte
   Tagalog (Phillipinisch): 3 Byre

6. BONUS-AUFGABE: In UTF-8 brauchen Buchstaben eine unterschiedliche Zahl von Bits. Wie würde eine optimale (im Hinblick auf den Speicherbedarf) Kodierung aussehen? Welche wichtige Eigenschaft muss eine solche Kodierung erfüllen? Entwerfen Sie eine optimale Kodierung für eine Sprache mit nur den 5 Buchstaben A,B,C,D,E, wobei A und E häufiger vorkommen als B,C, und D. (5 Punkte)

   Wie brauchen kurze Kodierungen für die häufige Buchstaben und längere für die wenige häufige. Das Problem, das dann entsteht ist, dass wir bei einer Folge von Bits nicht wissen, wo ein Bit aufhört und ein neues anfängt. In dem meisten Codepages, fängt ein neuer Code immer nach genau 8 Bits an. Das Problem wird dadurch gelöst, dass wir die Codes so wählen, dass kein einziger Code der Anfang eines anderen Codes ist.

   Eine mögliche Lösung:
   A: 00
   B: 10

C: 110
D: 111
E: 01