

# Übung zur Vorlesung “Einführung in die Computerlinguistik und Sprachtechnologie”

Wintersemester 2016/2017, Prof. Dr. Udo Hahn, Sven Büchel

Übungsblatt 5 vom 02.12.2016

Abgabe bis 08.12.2016, 23.59 Uhr; per Email (PDF-Format) an  
sven-eric.buechel@uni-jena.de

---

## Aufgabe 1 : Quantoren-Skopus

2 Punkte

Suchen Sie ein neues Beispiel für die Ambiguität bezüglich des Quantoren-Skopus und diskutieren Sie die möglichen Interpretationsvarianten dieses Beispiels.

**Lösung:** Alle Eichhörnchen haben eine Haselnuss.

- Alle Eichhörnchen besitzen die *gleiche* Haselnuss.
- Jedes Eichhörnchen besitzt eine *eigene* Haselnuss.

---

## Aufgabe 2 : Recherche

3 Punkte

a) ELIZA

Durch welche Funktion ist das Programm Eliza bekannt geworden? Nachahmung eines Psychotherapeuten.

b) Information Extraction und Information Retrieval

Erklären Sie **kurz** den Unterschied zwischen Information Extraction (IE) und Information Retrieval (IR).

**Lösung:** IE: Umwandlung von unstrukturierten Daten (Text) in strukturierte Daten entsprechend vorher festgelegter Templates. IR: Auffinden passender Dokumente bezüglich einer Suchanfrage (Query) aus einer Dokumentensammlung.

---

## Aufgabe 3 : Silbentrennung

5 Punkte

Definieren Sie eine Funktion, die geeignet ist eine vereinfachte Version der 1. Regel für Silbentrennung im Deutschen anzuwenden. Diese lautet (vereinfacht):

Es wird vor jedem Konsonanten getrennt, auf den ein Vokal folgt.

Betrachten Sie dabei die Liste `vokale` als gegeben, die alle Vokale (auch die Umlaute) des Deutschen als Strings enthält.

Ihre Funktion soll einen String entgegennehmen und alle Stellen finden, an denen laut dieser Regel getrennt werden könnte. Schreiben Sie dazu einfach an den passenden Stellen Trennstelle eintragen.

**Anmerkungen:**

- Sie können für diese Übung mit `String[n-1]` den n-ten Buchstaben eines Strings abfragen, also etwa für "Mama" [0] das "M".
- Zudem können Sie mit `length(String)` die Länge eines Strings abfragen, im obigen Beispiel also 4.
- Sie können mit `in` bzw. `not in` testen, ob etwas (nicht) in einer Liste ist. So würde etwa ein eingerückter Block unter `if "k" not in vokale` ausgeführt werden.
- Wörter können nicht vor dem ersten Buchstaben getrennt werden!

**Lösung:**

```
def trenne(↓wort)
  i ← 1
  while i+1 < length(wort)
    if wort[i] not in vokale
      if wort[i+1] in vokale
        Trennstelle eintragen
    i ← i + 1
```