

Übung zur Vorlesung “Computerlinguistik I”

Wintersemester 2017/2018, Prof. Dr. Udo Hahn, Sven Büchel

Übungsblatt 4 vom 23.11.2017

Abgabe bis 27.11.2017, 23.59 Uhr; per Email (PDF-Format) an sven.buechel@uni-jena.de

Aufgabe 1 : Ambiguität bezüglich PP-Anbindung (3 Punkte)

Gegeben sei noch einmal der Satz mit den acht Präpositionalphrasen (PP) aus den Einführungsfolien:

Die Tarifparteien haben sich [am frühen Mittwoch morgen]₁ [in Bad Nauheim]₂ [nach einem 16-stündigen Verhandlungsmarathon]₃ [auf einen Tarifabschluss]₄ [in Höhe]₅ [von 1,2 Prozent]₆ [für die 160.000 Beschäftigten]₇ [in der hessischen Bekleidungsindustrie]₈ geeinigt.

- Welche der oben stehenden PPs beziehen sich direkt auf das Prädikat und welche beziehen sich auf andere PPs? Geben Sie (zum Beispiel graphisch in Form eines Baumes) an, worauf sich jede der acht PPs bezieht, damit die korrekte Satzsemantik ausgedrückt wird. Ist es linguistisch plausibel, nur binäre Bäume bzw. Anbindungen zuzulassen?
-

Aufgabe 2 : Algorithmus “Vokalersetzung” (2,5 Punkte)

Unten sind drei Funktionen angegeben, von denen eine die beiden anderen aufruft. Lösen Sie folgende Aufgaben:

a) 0,5

Welchen Zweck erfüllt die Funktion `vokale`?

b) 0,5

Welchen Zweck erfüllt die Funktion `ersetzeVokale`?

c) 0,5

Welchen Zweck erfüllt die Funktion `aufrufer(wort)`?

d) 1

Gegeben seien die Wörter *Glas*, *Donaudampfschiffahrtskapitän*, *Schildkröte*, *Naturschutzpark*, *Uhu*, *jetzt*. Welche Ausgabe hat `aufrufer(wort)` für jedes dieser Wörter? Es reicht die veränderten Wörter anzugeben!

```
def vokale(↓wort, ↑counter)
    counter ← 0
    for buchstabe in wort
        if buchstabe ist ein Vokal
            counter ← counter + 1
```

```
def ersetzeVokale(↓wort, ↑wort)
    i ← 0
    while i < length(wort)
        if wort[i] ist ein Vokal
            wort[i] ← "u"
        i ← i + 1
```

```
def aufrufer(↓wort)
    if vokale(wort) > 1
        wort ← ersetzeVokale(wort)
    print wort
```

Aufgabe 3 : Rekursion (4,5 Punkte)

Definieren Sie eine rekursive Funktion, die für eine gegebene Liste `zahlen` genau die erste Zahl zurückgibt, die größer als 20 ist (z.B. `zahlen = [3, 1, 32, 5, 40, 7]` → Ausgabe 32). Falls die Liste keine entsprechende Zahl enthält, so soll 20 zurückgegeben werden.