

Übung zur Vorlesung “Einführung in die Computerlinguistik und Sprachtechnologie”

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

Übungsblatt 3 vom 18.11.2016

Abgabe bis 24.11.2016, 23.59 Uhr; per Email (PDF-Format) an

sven-eric.buechel@uni-jena.de

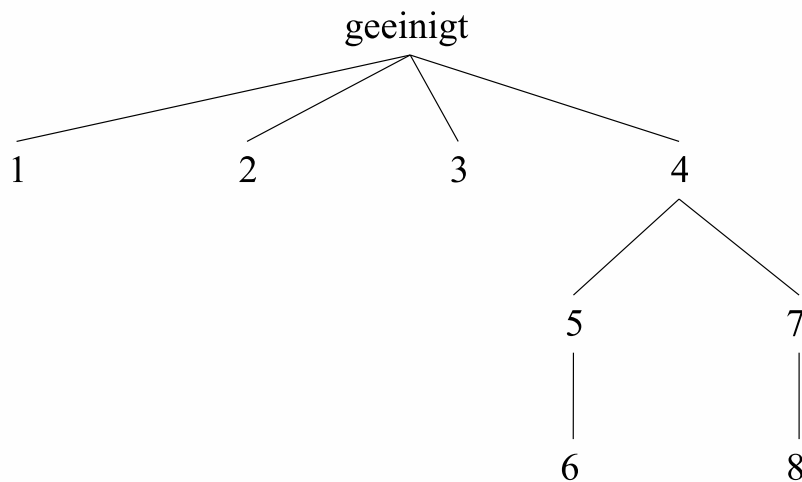
Aufgabe 1 : Ambiguität bezüglich PP-Anbindung (3 Punkte)

Gegeben sei noch einmal der Satz mit den acht Präpositionalphrasen (PP) auf Seite 53 der 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.

1. In welcher Anordnung müssen sich die Präpositionalphrasen in diesem Satz aufeinander beziehen, so dass die Satzsemantik ausgedrückt wird, die Sie für korrekt halten? Ist es linguistisch plausibel, nur binäre Bäume/Anbindungen zuzulassen?

Lösung:



Aufgabe 2 : Algorithmen (7 Punkte)

Für die folgenden Übungsaufgaben gilt, dass bei

$String1 - String2 = String2'$

$String1$ linksbündig aus $String2$ zu entfernen ist, vgl. das folgende Beispiel:

$aa - aaab = ab$

Zudem können Sie für Bedingungen fragen, ob ein String länger/kürzer als ein anderer ist und womit er beginnt, z.B. für einen gegebenen String a :

```
if a beginnt mit "foo"
    if a laenger als "a"
        print "bar"
```

a) Verstehen I (1.5 Punkte)

Was gibt die folgende Funktion für die Eingaben ("a", "aab") und ("foo", "f") zurück?

```
def foo(↓a, ↓b, ↑c)
  if a laenger als b
    c ← b - a
  else
    if b laenger als a
      c ← a - b
    else
      c ← a
```

Lösung: "ab" und "oo"

b) Verstehen II (1.5 Punkte)

Was gibt die folgende Funktion für die Eingaben ("a", "bba") und ("foo", "f") zurück?

```
def foo(↓a, ↓b, ↑c)
  if a beginnt mit b
    c ← b - a
  else
    c ← a + b
```

Lösung: "abba" und "oo"

c) Iterativ (4 Punkte)

Definieren Sie eine Funktion, die von einer Eingabe e linksbündig alle initialen "a" entfernt und das Ergebnis als r zurückgibt. Wir entfernen dadurch die Präfixe a,aa,aaa,... und machen etwa "aabab" zu "bab". Formulieren Sie die Funktion iterativ (mit einer Schleife).

Lösung:

```
def foo(↓e, ↑r)
  r ← e
  while r beginnt mit 'a'
    r ← 'a' - r
```