

# Übung 05: Musterlösung

Programmiertechniken in der Computerlinguistik II, FS 16

27. Mai 2016

## 1 Längste gemeinsame Teilfolge

a) top-down Ansatz:

LCS Matrix

	0	C	O	S	M	I	C
0	∅		∅	∅			
C		C=C, [C]	C!=O, [C]	C!=S, [C]			
O	∅	O!=C, [C]	O=O, [C,O]	O!=S, [C,O]			
M	∅	M!=C, [C]	M!=O, [C,O]	M!=S, [C,O]	M=M, [C,O,M]		
I	∅	I!=C, [C]	I!=O, [C,O]	I!=S, [C,O]	I!=M, [C,O,M]	I=I, [C,O,M,I]	
C		C=C, [C,C]	C!=O, [C,C] or [C,O]	C!=S, [C,C] or [C,O]	C!=M, [C,O,M]	C!=I, [C,O,M,I]	C=C, [C,O,M,I,C]
S				S=S, [C,O,S]	S=M, [C,O,S] or [C,O,M]	S!=I, [C,O,M,I]	S!=C, [C,O,M,I,C]

- b)  $O(n * m)$ . Die längste anzunehmende Laufzeit würde eintreffen, falls die längste gemeinsame Teilfolge die Länge 0 hat und somit die ganze Matrix gefüllt wird.
- c) Dieser Lösungsansatz ist dynamisch, da aus einem Problem kleinere Unterprobleme gebildet werden, die Lösungen zu den Unterproblemen gespeichert werden und deshalb bei einer erneuten Iteration nicht noch einmal dasselbe Unterproblem gelöst werden muss.

## 2 Levenshtein-Distanz

- a)
- Satz A: Computerlinguistik 2 ist spannend.
  - Satz B: Computerlinguistik macht Spass und ist spannend!
  - **3.0** (2 entfernen) + **3.0** (*macht* einfügen) + **3.0** (*Spass* einfügen) + **3.0** (*und* einfügen) + **0.1** (. mit ! ersetzen) = **12.1**

Die Lösung ist nicht 22.1, da man die 2 entfernen und *macht* einfügen kann, anstatt die 2 mit *macht* zu ersetzen. Somit verwendet man zwar **zwei Operationen**, welche je **3.0** kosten, vermeidet dafür eine Operation, welche **16.0** kostet.

b) siehe Skript

c) siehe Skript

- d) Der Satz im Brown Korpus mit der kleinsten generellen Bearbeitungsdistanz zu *I thought you loved me.* ist *I wish you was Henry.* mit der Distanz **3.9** (3x ersetzen).