

Zusammenfassung zur Vorlesung Basismodul Computerlinguistik

Nützliche Python-Module & Levenshtein-Automaten

13.01.2022

- 1 Nützliche Python-Module (Helfen beim Abschlussprojekt)
 - pickle
 - argparse
- 2 Tafelübung: Levenshtein-Automaten

- Ermöglicht komfortable (De-)Serialisierung von Python-Objekten (wie Liste, Dictionary usw. auch user-defined)
Serialisierung: Konvertierung eines Objekts in einen Bytestream
- Speichert Daten (Python-Objekte) als Bytestream in einer Datei
→ nicht als Textdatei lesbar (vs. JSON)
- Beispiel

- übernimmt das Parsen von Kommandozeilen-Argumenten
- eine einfache Variante: `sys.argv`
- Beispiel

Features von argparse

- Ein Vorteil: argparse erstellt automatisch eine Hilfe-Funktion
- Typisierung der Argumente (int / str / bool)
- Beschreibungstexte für das Programm und die Argumente
- Neben positionalen Argumenten auch Optionale möglich

Tafelübung: Levenshtein-Automaten

Aufgabe Es sei v das Pattern *haus* und w das Input-Wort *baum*. Die Schranke für den Levenshtein-Abstand ist 2.

- 1 Wie sieht die Folge der Bitvektoren aus, die als Automaten-Eingabe zu verwenden ist, um zu testen, ob der Levenshtein-Abstand zwischen v und w kleinergleich 2 ist.
- 2 Zeige mittels des Dreiecksbereichs, ob der Levenshtein-Abstand zwischen v und w kleinergleich 2 ist.

① pattern $v = \text{haus}$

input word $w = \text{baum}$

Schranke $b = 2$

max. Länge des Bitvektors:

$$L = 2b + 2 = 6$$

haus baum

$$\chi(b, \text{haus}) = 000000$$

6

$$\chi(a, \text{haus}) = 00100$$

5

$$\chi(u, \text{haus}) = 0010$$

4

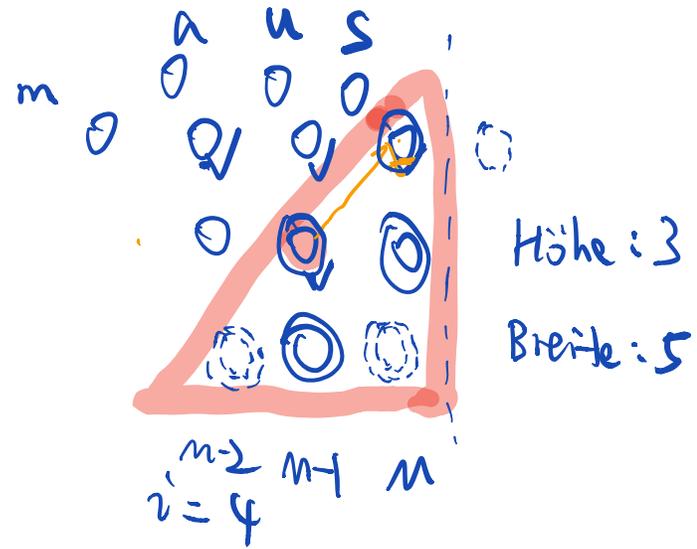
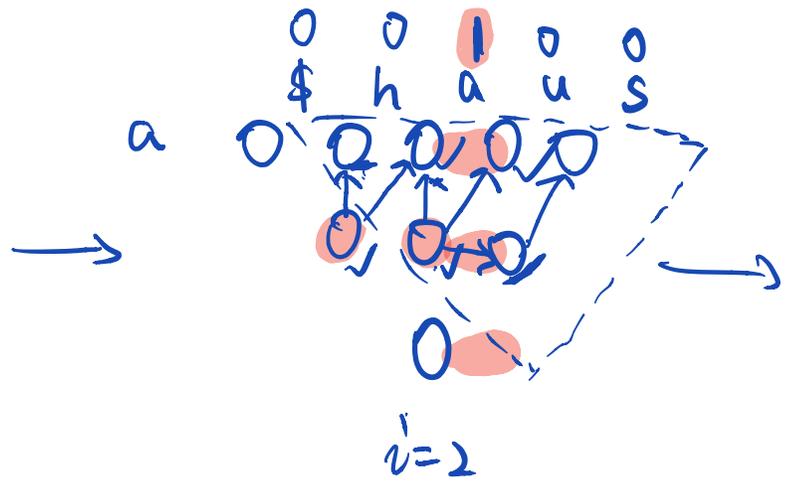
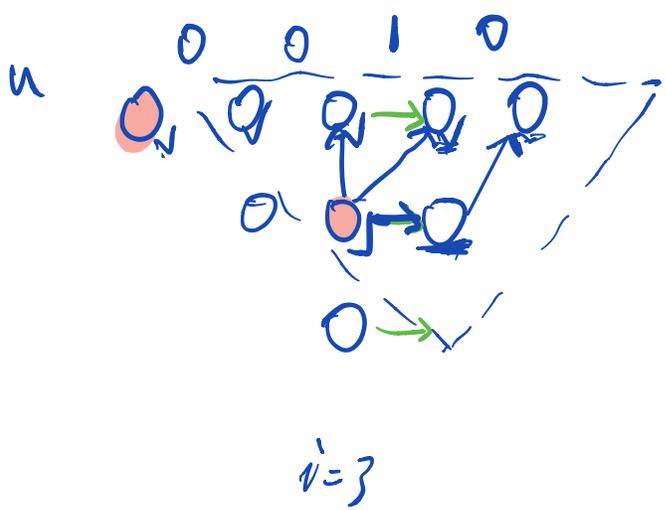
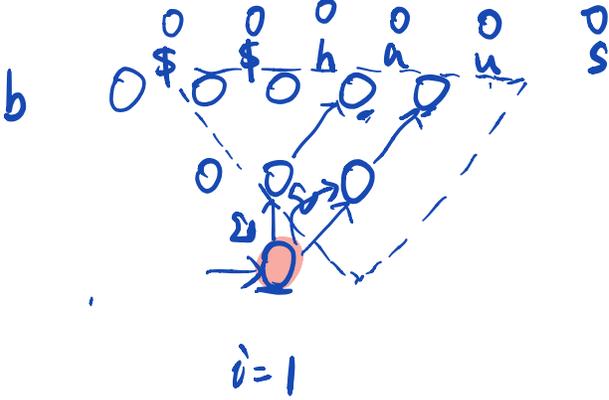
$$\chi(m, \text{aus}) = 000$$

3

② Dreiecksbereich

Höhe: $b+1 = 3$

Breite: $2b+1 = 5$



Finalzustand, aktiv ✓

$v, w < 2$.