Elternannotation

In dieser Übung schreiben Sie ein Programm, welches einen Parsebaum pro Zeile in Klammer-Notation aus einer Datei einliest und eine Elternannotation ausführt. Hier ist ein Beispiel:

Eingabe: (S (NP (DT This)) (VP (VBZ is) (NP (DT a) (NN sentence))))

Ausgabe: (S-NONE (NP-S (DT-NP This)) (VP-S (VBZ-VP is) (NP-VP (DT-NP a) (NN-NP sentence))))

Der Wurzelknoten erhält die Elternkategorie `NONE`.

Das Programm soll zunächst den Parsebaum in eine interne Baumstruktur einlesen. Dann soll der Parse rekursiv durchwandert und mit Elternannotationen ausgegeben werden.

Programmaufruf: `python annotate.py parses.txt`

Verwenden Sie bei dieser Aufgabe objektorientierte Programmierung mit einer Klasse `Node`. Das Hauptprogramm sollte so aussehen:

```
with open(sys.argv[1]) as file:
    for line in file:
        ...
        root = Node(line)
        ...
        print(root)
```

Das Einlesen des Parsebaumes übernimmt also die init-Methode der Klasse *Node*. Sie arbeitet dabei rekursiv und ruft sich selbst auf mit `child = Node(input_string)`.

Verwenden Sie zum Einlesen des Parsebaumes wieder die Methode des Parsens mit rekursivem Abstieg, bei der Sie die Zeichen ohne Vorausschauen strikt von links nach rechts verarbeiten.

Wenn Sie einen *Formatfehler* in der Eingabe erkennen, lösen Sie wieder eine Exception aus und machen mit der nächsten Eingabezeile weiter (wie in der Übung zum Parsen mit rekursivem Abstieg).

Für die Ausgabe schreiben Sie eine Methode `__str__(self)`, welche den Parsebaum rekursiv wieder in einen String umwandelt und zurückgibt.

Vorüberlegungen

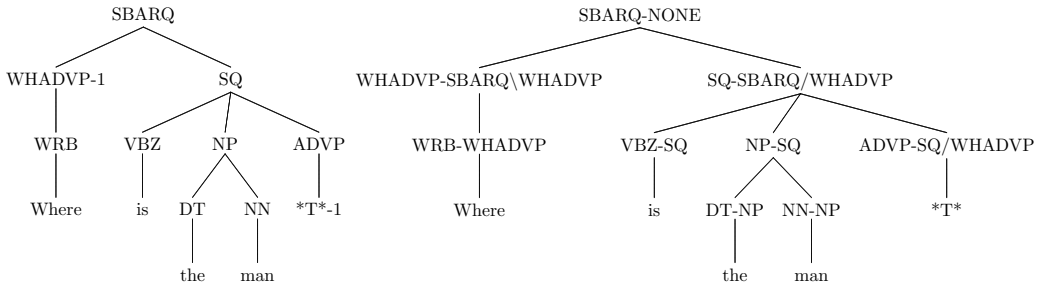
- Schreiben Sie eine allgemeine Meta-Grammatik, welche die Parsebäume von beliebigen Grammatiken im obigen Klammerformat generieren kann. (3 Regeln und 2 Nichtterminale genügen.)
- Welche Attribute sollte die Klasse `Node` besitzen?
- Wie gehen Sie mit Leerzeichen um?

Spurmerkmale

Erweitern Sie die Parsebaum-Annotation um Spurmerkmale. Es sollen alle Knoten auf dem Weg zwischen Spur (bspw. *T*-1) und Füller (bspw. WHADVP-1) mit der Füllerkategorie annotiert[1] werden (außer dem Knoten, der Füller und Spur dominiert). Hier ist ein Beispiel:

Eingabe: (SBARQ (WHADVP-1 (WRB Where)) (SQ (VBZ is) (NP (DT the) (NN man)) (ADVP *T*-1)))

Ausgabe: (SBARQ-NONE (WHADVP-SBARQ\WHADVP (WRB-WHADVP Where)) (SQ-SBARQ/WHADVP (VBZ-SQ is) (NP-SQ (DT-NP the) (NN-NP man)) (ADVP-SQ/WHADVP *T*)))



Ein Spurknoten ist ein Terminalknoten, der mit '-n' am Ende annotiert ist, wobei n eine beliebige Zahl ist und vor dem Bindestrich "-" ein "*" steht. Ein Füllerknoten ist ein nichtterminaler Knoten, der mit '-n' annotiert ist. Für jede Spur gibt es genau einen passenden Füller mit demselben Index und umgekehrt. Es können mehrere Spur-Füller-Paare mit unterschiedlichen Indizes in einem Satz auftauchen, deren Pfade sich überlappen.

Trennen Sie jeden Spur-Index (im Beispiel: -1) beim Einlesen des Parsebaumes (mit einem regulären Ausdruck) vom Kategorienamen ab und speichern Sie ihn in einem Attribut. Den komplexen neuen Kategorienamen setzen Sie erst bei der Ausgabe des Parsebaumes zusammen. Für die Berechnung der Spurmerkmale schreiben Sie die folgenden drei Methoden:

- `add_trace_features`: durchwandert den Parsebaum top-down und rekursiv auf der Suche nach Füllern. Wenn ein Füller gefunden wurde, wird die Methode `annotate_trace_up` aufgerufen.
- `annotate_trace_up`: Diese Methode wandert im Parsebaum nach oben, weist den besuchten Knoten Spurmerkmale zu und ruft `annotate_trace_down` für alle Geschwisterknoten auf, um top-down nach einer passenden Spur zu suchen. Wenn eine solche Spur gefunden wurde, wird `annotate_trace_up` beendet.
- `annotate_trace_down`: Diese Methode duchwandert rekursiv den Teilbaum unter dem aktuellen Knoten und sucht nach einer passenden Spur zu dem gefundenen Füller. Die Methode liefert den Wert *True* zurück, wenn eine Spur gefunden wurde, sonst *False*. Wenn eine Spur gefunden wurde, wird dem Knoten ein Spurmerkmal zugewiesen.

[1] Diese Annotation für Spuren wird in der englischen Penn Treebank verwendet.

Versuchen Sie, möglichst einfachen und kurzen Code zu schreiben. Bei einer sehr guten Implementierung reichen 100 Zeilen Code (ohne Kommentare).

Vorüberlegungen

- Spielen Sie an einem Beispielparse durch, wie die drei Methoden die Spurannotation durchführen.
- Welche zusätzlichen Node-Attribute benötigen Sie? Welche Argumente müssen Sie den Methoden übergeben?

Schicken Sie das Programm an `schmid@cis.lmu.de`.