Reg-was?

Das System der regulären Ausdrücke (Englisch: *regular expressions*) ist ein Standard zum Beschreiben von Zeichenketten und textuellen Bausteinen. Die meisten Texteditoren lassen neben der normalen Textdurchsuchungsfunktion auch eine erweiterte Suche mithilfe regulärer Ausdrücke zu (Zum Aktivieren im Suchfenster (meist aufzurufen durch Strg+F, in Textpad durch F5) "Reguläre Ausdrücke", "Regex", "Regular expressions", o.ä. markieren). Auch Programmierstprachen erlauben es, reguläre Ausdrücke zu integrieren und Textarbeit auf diese Weise zu automatisieren, was sie auch in der Linguistik zu einem überaus praktischen Hilfsmittel macht. Wie aus dem vorausgehenden Kapitel hervorgegangen sein sollte, lassen sich Zeichen im Hinblick auf Typ, Position und Kombination beschreiben. Zeichenketten liegen Muster zugrunde, die formalisiert und dazu herangezogen werden können, um dieselben abstrakten Muster auch hinter anderen Zeichenketten zu finden.

Auf der Suche nach den abstrakten Mustern

Doch wie entnimmt man Textdaten diese abstrakten Muster? Im vorausgehenden Kapitel wurde das Muster hinter den drei Sätzen *Alle Auen sind grün.*, *Auch Autobahnen sind grau.* und *Mein Hund kann Sitz.* als "Der Satz besteht aus vier Wörtern, von denen sich das erste, das dritte und das vierte aus je vier Buchstaben zusammensetzen. Das zweite ist beliebig lang." umschrieben. Mit einem Formalismus, der genau diese Definition beinhaltet, ließen sich alle Sätze eines Textes, auf welche dieses Muster zutrifft, herausfiltern. Ist das nun DAS zugrundeliegende abstrakte Muster der drei Sätze? Nein, es ist EIN mögliches Muster. Ein aufmerksamerer Blick auf die drei Datensätze wird weitere Gemeinsamkeiten aufdecken. Zum Beispiel ist der vorletzte Buchstabe des dritten Wortes in allen drei Sätzen ein *n*, die zweiten Wörter enthalten zumindest alle ein *n*, die ersten Buchstaben der ersten beiden Wörter sind stets großgeschrieben, ... Mit anderen Worten: Vielleicht ist für uns gar nicht von Interesse, wie viele Buchstaben und Wörter die Sätze Enthalten, und wir bemühen uns vielmehr darum, Sätze zu finden, an deren zweiter Stelle sich ein Nomen (folglich großgeschrieben) befindet. Oder wir wollen lediglich einen Datenausschnitt finden, in dem der Text durch keine Ziffern unterbrochen wird, oder auch nur eine Auflistung der Großbuchstaben, die in den Daten vorkommen, erstellen, oder oder oder.
Das heißt? Ganz einfach: Die Muster (oder: Patterns) werden je nach gewünschtem Suchergebnis von jedem Suchenden selbst formuliert.

Zwischen "abstrakt" und "konkret"

Wie funktioniert das nun mit den Regulären Ausdrücken? Wenn wir in einem Regex-kompatiblen Texteditor unsere Suchanfrage formulieren, dann haben wir zum einen die Möglichkeit, dies mithilfe einer konkreten Zeichenkette zu tun (z.B. *Autobahnen*). So weit nichts Neues! Reguläre Ausdrücke erlauben es uns zudem, völlig abstrakte Suchanfragen zu formulieren, in denen kein einziger Buchstabe vorkommt (z.B. das letzte Zeichen am Ende einer jeden Zeile). Meist bewegt man sich jedoch dazwischen. Die Syntax regulärer Ausdrücke ermöglicht es, konkrete Zeichen (z.B. *A*, *b*, *1*, *!*, etc.) und sogenannte Metazeichen (z.B. Platzhalter, Wiederholung, Alternativen, Position) zu kombinieren, um so eine möglichst präzise und nötigenfalls komplexe Suchanfrage zusammenzustellen. Auch Metazeichen müssen in dem Suchstring auf irgendeine Weise ausgedrückt werden, weshalb einige konkrete Zeichen zu Metazeichen umgewidmet wurden (z.B. *.*, *{ }*, *[ ]*, *|*, etc.). Ein häufiger Anfängerfehler besteht in der Tat darin, dass bestimmte Zeichen eigentlich konkret gemeint sind, bei der Durchführung der Suche jedoch als Metazeichen erkannt werden.

Previous Vorab Up Reguläre Ausdrücke in der Arbeit mit Textdaten Von null bis unendlich Next