Ersatz gefunden

Die Klammern erweisen sich als überaus hilfreich, arbeitet man mit der Ersetzfunktion eines Texteditors (Die Funktion "Ersetzen" befindet sich meist irgendwo in der Nähe der Funktion Suchen "Suchen". Bei Notepad/ Textpad sind beide unter dem Menüpunkt "Suchen"/ "Search" gelistet). Diese Funktion ist insofern sinnvoll, als dass sie die Möglichkeit eröffnet, Textdaten und ihm zugrundeliegende Muster aktiv zu manipulieren.

Eine einfache Ersetz-Operation bestünde beispielsweise darin, alle Formen des Farbadjektivs *rosso* in einem Text durch die Nennform *rosso* zu ersetzen. Ein regulärer Ausdruck zum Auffinden aller Formen könnte dabei so lauten:

Ausdruck	**matcht...**
ross[oaie]	rosso, rossa, rossi, rosse

Im "Ersetzen durch"-Feld wird nun die überschreibende Form definiert, hier *rosso*.

Ausdruck	**matcht...**	**Ersetzen durch...**	**Ergebnis**
ross[oaie]	rosso, rossa, rossi, rosse	rosso	rosso, rosso, rosso, rosso

Eine weitere Aufgabe wäre zum Beispiel das Ersetzen der Formen von *rosso* durch entsprechende Formen von *giallo* (sprich: *giallo*, *gialla*, *gialli*, *gialle*). Auch diese lässt sich einfach lösen, indem man den Endvokal im regulären Ausdruck ignoriert und nur das lexikalische Morphem ersetzt. Alternativ lässt sich jedoch auch mit den runden Klammern arbeiten:

Ausdruck	**matcht...**	**Ersetzen durch...**	**Ergebnis**
ross	ross, ross, ross, ross	giall	giallo, gialla, gialli, gialle
ross([oaie]) oder (z.B. in älteren Textpad-Versionen): ross\([oaie]\)	rosso, rossa, rossi, rosse	giall\1	giallo, gialla, gialli, gialle

Während die erste Lösung wahrscheinlich einleuchtet, bedarf die zweite einer weitergehenden Erklärung: Allen Inhalten, die sich in runden Klammern befinden, wird im Rahnen eines Ersetzvorgangs eine Kennzahl zugeordnet. Der Inhalt der ersten Klammer erhält die Kennzahl 1, der der zweiten 2, der der dritten 3 und so weiter. Sind die Klammern geschachtelt, beginnt der Zähler bei bei der äußersten linken Klammer und zählt von links nach rechts jede öffnende Klammer durch.

In dem vorliegenden Beispiel besteht der Inhalt entweder aus dem Buchstaben *o*, *a*, *i* oder *e*. Dieser Inhalt kann nun als Zeichen (bzw. Zeichenkette) in die ersetzende Form mithilfe der Kombination von Backslash und der entsprechenden Kennzahl übernommen werden. Dies funktioniert folgendermaßen. Der Ausdruck *ross([oaie])* sucht nach einer Zeichenfolge, auf die das Muster *ross[oaie]* passt, trifft zuerst auf *rossa* und setzt *\1* mit *a* gleich. Nun wird *rossa* durch giall gefolgt von *\1* (also *a*) ersetzt. Der Ausdruck sucht nun nach dem nächsten Match, z.B. *rossi*. Auch hier wird *i* mit *\1* gleichgesetzt, bevor *rossi* durch *giall\1* ersetzt wird. Was deutlich werden soll, ist, dass der Backslash mit der Kennzahl keinen Platzhalter übernehmen, sondern das jeweils konkrete Zeichen. Dies liegt darin, dass die Ersetzen-Funktion einen konkreten Output generiert. Die meisten Metazeichen wie Positionsmarker und Platzhalter sind in der "Ersetze durch"-Zeile folglich nicht zulässig und werden als konkrete Zeichen interpretiert. Zu den zugelassen Metazeichen zählen die bereits präsentierten **Verweise** mithilfe von Kennzahlen. Auch kann auf die gesamte im Suchfenster definierte Zeichenfolge verwiesen werden. Das im Ersattfenster platzierte Metazeichen *&* steht repräsentierend für den gesamten im Suchfenster formulierten Ausdruck, bzw. jede Zeichenfolge, mit der der Ausdruck matcht. *&* lässt sich sinnvoll einsetzen, wenn man einer Zeichenfolge, die in ihrer Form erhalten werden soll, weitere Informationen o.ä. hinzufügen will. Zum Beispiel kann man alle Adjektive mit der Pattern ross[oaie] durch *"&"* in Anführungsstriche setzen. Alle durch *ross([oaie])* definierten Zeichenfolgen lassen sich durch *& e giall\1* ersetzen - mit der Folge, dass alle Vorkommen von *rosso/a/i/e* zu *rosso/a/i/e e giallo/a/i/e* werden.

Zugelassen sind zudem Metazeichen für einige **Steuerzeichen**, d.h. Zeichen, die vorhanden, aber nicht sichtbar sind. Dazu zählen u.a. die Shorthand-Expressions:

Ausdruck	**Ergebnis**
\n	Fügt eine neue Zeile ein
\t	Fügt einen Tabulator ein

\1\n\2 trennt zwei im Suchfenster definierte Gruppen durch einen Zeilenumbruch (z.B. *(a)(b)*, mit dem Ergebnis: *ab -> a nächste Zeile b*), *\t* durch einen Tabulator.

Zudem bietet die Ersetzfunktion die Möglichkeit, Klein- und Großschreibung zu erzwingen. Dafür stehende folgende Shorthands zur Verfügung:

Shorthand	bewirkt...
\u	Das erste der nachfolgenden konkreten Zeichen wird großgeschrieben.
\U	Alle nachfolgenden konkreten Zeichen werden großgeschrieben.
\l	Das erste der nachfolgenden konkreten Zeichen wird kleingeschrieben.
\L	Alle nachfolgenden konkreten Zeichen werden kleingeschrieben.

Nehmen wir zur Veranschaulichung eine Textdatei, die aus der Buchstabenfolge *abc* zusammengesetzt ist. Der Ersetzausdruck *\u&* (entspricht: "Erzwungene Kleinschreibung des ersten Buchstabens, Suchstring übernommen") würde diese zu *Abc* umwandeln, *\U* zu *ABC*, \U&def zu ABCDEF. Genauso funktioniert es, will man die Kleinschreibung von Großbuchstaben erzwingen.

Was soll ich damit?

Keine Idee für Anwendungsmöglichkeiten? Eventuell erfordern Tools, so wie das Annotationstool TreeTagger, dass Texte in einer bestimmten Form vorliegen. Bei TreeTagger entpräche dies der Formatierung *ein Wort pro Zeile*. Eine solche Form könnte zum Beispiel hergestellt werden, indem alle Leerstellen ( ) durch *\n* ersetzt werden. Auch lassen sich Daten von überflüssigen Zeichen reinigen. Will ich beispielsweise Interpunktionszeichen aus meinem Korpus entfernen, so ist dies über eine Ersetzung von *[.?:,!]* durch *nichts* (also durch ein leeres Ersetz-Feld) zu bewerkstelligen. Auch die Anordnung von Textbausteinen lässt sich durch die Ersetzfunktion mithilfe der Verweise verändern. Dazu werden Gruppen definiert (z.B. *(a)(b)(c)*) und ihre Reihenfolge unter Rückgriff auf ihre Kennzahlen verändert (z.B. zu *\3\2\1*).

Previous Und oder

Up Reguläre Ausdrücke in der Arbeit mit Textdaten