durch die Spielernummer und die Spielzeit (von Minute, bis Minute), die Aufstellung

- Die Auswechslungen, wobei jeweils angegeben ist, welcher Spieler gegen welchen anderes Spieler in welcher Minute getauscht wurde und schließlich
- den Toren der Begegnung, ergänzt durch die Minute, den Schützen und wer das Tor geschossen hat.

In einem ersten Schritt werden die Angaben zum Spiel in alle diese Einzelteile zerlegt und die **Aufstellung** und **Tore** noch als eine Zeichenkette beisammen gelassen aber die Angaben zum Spiel bereits in die Felder **Datum**, **Gegner**, **Ergebnis**, **Halbzeitergebnis**, **Schiedsrichter**, **Zuschauerzahl**, **Spielort** usw. aufgeteilt.

Die **Aufstellung** und die **Tore** bleiben zunächst als Text in einem genügend großen Textfeld. 255 Zeichen (die Vorgabe *„Kurzer Text“* in Microsoft Access) genügt nicht, man muss *„Langer Text“* einstellen, sonst werden bei einigen Spielen Textteile abgeschnitten. In diesem ersten Schritt entsteht also die einzige Tabelle **Spiel**.

Dazu muss aber die Textdatei gründlich umformatiert werden, damit ein lesendes Programm einfach erkennen kann, wann ein Spiel beginnt, wann es endet und welche Abschnitte daraus in welches Feld kommen sollen.

Das folgende Beispiel zeigt am Spiel 716 gegen die Elfenbeinküste, das auf der vorigen Seite auch im Original zu sehen ist, wie dieser maschinenlesbare Text schließlich ausschaut. Zur Verbesserung der Lesbarkeit wurden die Feldnamen fett gedruckt.

####
***#Begegnung#**Österreich:Elfenbeinküste*
***#Spiel#**716*
***#Ergebnis#**0:3*
***#Pause#**0:1*
***#Art#**Freundschaftsspiel,*
***#Datum#**14.November 2012*
***#Details#**Linz, Linzer Stadion, 13.832 Zuschauer,*
***#Schiedsrichter#**Kralovec (CZE)*
***#Aufstellung#**(GK 24) Hein4 Lindner/FK Austria Wien – (2) György Garics/FC Bologna, (3) Aleksandar Dragovic/FC Basel, (4) Emanuel Pogatetz/VfL Wolfsburg, (13) Markus Suttner/FK Austria Wien – (7) Marko Arnautovic/SV Werder Bremen, (18) Christoph Leitgeb/FC RB Salzburg, (8) David Alaba/FC Bayern München, (16) Jakob Jantscher/Dynamo Moskau – (6) Andreas Ivanschitz/1.FSV Mainz 05 – (21) Marc Janko/Trabzonspor*
***#Austausch#**(17) Florian Klein/FC RB Salzburg für Garics (46.), (15) Sebastian Prödl/SV Werder Bremen für Pogatetz (46.), (14) Julian Baumgartlinger/1.FSV Mainz 05 für Leitgeb (46.), (19) Veli Kavlak/Besiktas JK für Alaba (59.), (9) Andreas Weimann/Aston Villa FC für Ivanschitz (64.), (11) Martin Harnik/VfB Stuttgart für Jantscher (76.)*
***#Teamchef#**Marcel Koller*
***#AufstellungG#**Boubacar Barry/KSC Lokeren, Igor Lolo/Kuban Krasnodar, Kolo Touré/Manchester City, Sol Bamba/Trabzonspor, Arthur Boka/VfB Stuttgart – Romaric/Real Saragossa, Arouna Kone/Wigan Athletic, Cheick Tioté/Newcastle United, Didier Ya Konan/Hannover 96, Max Gradel/AS Saint-Étienne – Wilfried Bony/Vitesse Arnheim*
***#Austausch#**Ismael Traore/Stade Brestois für Bamba (46.), Lacina Traoré/Anzhi Makhachkala für Bony (46.), Abdul Razak/Manchester City für Tiote (53.), Didier Drogba/Shanghai Shenhua für A. Kone (59.), Yaya Touré/Manchester City für Romaric, Salomon Kalou/Lille OSC für Ya Konan*
***#Teamchef#**Sabri Lamouchi*
***#Tore#**0:1 Ya Konan (44.) 0:2 Drogba (52.) 0:3 L. Traore (76.)*

Die gewählte Strategie

Jedes Feld in der späteren Tabelle **Spiel** ist eine Zeile in der Textdatei. Jede dieser Zeilen wird durch den späteren Feldnamen ***#Name#*** eingeleitet. Zwischen Spielen steht die Zeile *####*. Die Felder **Aufstellung**, **Tore** und **Austausch** sind vorläufig noch unstrukturierte Texte, die später, wenn aus diesem Text bereits die Tabelle **Spiel** geworden ist, per Programm in weitere Tabellen umgewandelt werden.

Wie formt man nun die Angaben des ÖFB zu den Spielen in die gezeigte Form mit den Rautezeichen um? Händisch wäre das ein mühsamer Weg. Man kann sich dafür aber hervorragende Hilfe in Form der ***Regular Expressions*** holen, die in Editoren für Programmier-Aufgaben und sogar in Word enthalten sind. Für diese Aufgabe verwende ich gerne den kostenlosen Editor ***Notepad++***. Im Dialogfeld von Notepad++ kann man einstellen, ob der Suchbegriff als ein ***Regulärer Ausdruck*** interpretiert werden soll.

Regular Expressions

Ein einfaches Suchen und Ersetzen erlaubt nur die Suche nach einem konkreten Text, also zum Beispiel nach *„Länderspiel“* oder *„Österreich“* aber nicht nach einem beliebigen Text, der im Zusammenhang mit *„Länderspiel“* vorkommt, zum Beispiel *„25. Länderspiel“* oder *„Österreich-Ungarn“*. Das *„25.“* und *„Ungarn“* ist bei jedem Spiel anders und daher versagt die einfache Suche.

Hier kommen die ***„Regular Expressions“*** ins Spiel. Grundsätzlich würde ein normaler Text in einer ***Regular Expression*** ebenso behandelt werden wie in einer normalen Suche.

Bestimmte Sonderzeichen haben aber eine besondere Bedeutung. Diese Zeichen sind:

.:()[]^+*?-\|

Daher kann man nach diesen Zeichen nicht direkt suchen sondern nur, indem man ihnen einem Backslash voranstellt. Um also zum Beispiel nach einem Plus-Zeichen zu suchen, verwendet man \+.

Einfache Umformungen

Einfach sind Umformungen wie *„Tore:…“* im Originaltext, denn es genügt nach *„Tore:“* zu suchen und durch *„#Tore#“* zu ersetzen. Dazu gehören auch Schreibfehler, wie das erwähnte *„Polser“* statt *„Polster“*.

Alles, was man jetzt korrigieren kann, kann man einheitlich in der ganzen Datei ausführen. Zum Beispiel die Entfernung mehrfacher Spaces, mehrfacher Zeilenumbrüche. Meist muss man diese Kleinigkeiten auch am Ende der Umformungen noch einmal ausführen, weil durch die Umformung wieder mehrfache Spaces oder Zeilenumbrüche entstehen.

Hier unterscheiden sich die Regular Expressions nicht von der normalen Suche.

Wie nun diese Spezialzeichen zu interpretieren sind, zeigen die folgenden Beispiele und auch die Sonderseite über Regular Expressions.

Der gesamte Prozess dieser Umwandlung ist hier nicht exakt darstellbar. Es gibt sehr viele Ausnahmen, hervorgerufen durch Tippfehler (manchmal steht irgendwo ein Abstand, dann fehlt er) oder durch verschiedenartigen Aufbau der Zeilen, wie man im Vergleich des ersten und des letzten Spiels sieht. Die Herstellung der strukturierten Datei ist daher eine Mischung aus systematischen Umformungen und manuellen Korrekturen.

Begegnung

Die Begegnung steht meist an erster Stelle eines Spiels und wird von keinem Text begleitet, daher muss man sich an dem *„Österreich“* orientieren. Aus *„Ungarn-Österreich“* oder *„Österreich-Ungarn“* soll daher werden *#Begegnung#Ungarn-Österreich* oder *#Begegnung#Österreich-Ungarn*, unabhängig davon, welchen Gegner Österreich hat. Solche Ersetzungen sind nun mit den Mitteln des einfachen *„Suchen und Ersetzen“* nicht mehr durchführbar und man benötigt die Hilfe der ***Regular Expressions***.

In dem Beispiel suchen wir nach einem Wort, das wir unverändert auch nach dem Ersetzungsvorgang verwenden wollen. Dafür verwendet man in der Regex-Sprache als Suchbegriff die Suche

(\w+)\-(Österreich)

und ersetzt durch

#Begegnung#\1\2\r\n#Gegner#\2\r\n#Heim#A\r\n

Diese „Formeln“ wollen erklärt werden:

(\w+)\-(Österreich)

() Eingeklammerte Begriffe werden in der Reihenfolge ihres Auftretens als Variable betrachtet, die als **\1**, **\2**… im Ersetzungsbegriff verwendet werden können.

\w Steht für irgendein alphanumerisches Zeichen, Pluszeichen **+** bedeutet, dass das Zeichen davon beliebig oft, mindestens aber einmal vorkommt. Das **\w+** ist eingeklammert, daher werden alle dadurch gefundenen Zeichen in die Variable **\1** gespeichert. Nach dem Bindestrich kann nicht direkt gesucht werden, daher muss ihm in der Suche ein Backslash vorangestellt werden.

####\r\n#Begegnung#\1\2\r\n#Gegner#\1\r\n#Heim#A\r\n

\1 steht für den Gegner und **\2** für „Österreich“. **####**, **#Begegnung#**, **#Gegner#** und **#Heim#A** werden unverändert ausgegeben. **\r** steht für das Steuerzeichen CR und **\n** für LF.

Durch diese Ersetzung entstehen die vier Felder **####**, **Begegnung**, **Gegner** und **Heim**. Der Zeilenrest nach den #...#-Feldbezeichnungen ist der eigentliche Feldinhalt und wird in die Datenbanktabelle **Spiel** importiert. Die Extra-Zeile **####** ist die Trennung zwischen zwei Länderspielen, an der sich das spätere Einleseprogramm orientieren kann.

Fassen wir also zusammen. Durch diese Ersetzung werden aus der Zeile

Ungarn-Österreich

die Zeilen

####
#Begegnung#Ungarn-Österreich
#Gegner#Ungarn
#Heim#A

D.h. etwa die Hälfte der Spiele wird auf diese Weise umgeformt.

Diese Suche muss noch einmal durchgeführt werden und zwar für den Fall von Heimspielen, weil dort die beiden Gegner ihren Platz tauschen. Die Such- und Ersetzungkommandos lauten dann:

Suchen
(Österreich)\-(\w+)
Ersetzen
####\r\n#Begegnung#\1\2\r\n#Gegner#\2\r\n#Heim#H\r\n

Grundsätzlich könnte man mit den Mitteln der Regular Expressions sogar beide Ersetzungen in einem einzigen Vorgang durchführen, doch lohnt sich in diesem Projekt dieser zusätzliche