

# Reguläre Ausdrücke

*Jeffrey E. E Eriedl*

*Deutsche Übersetzung von  
Andreas Karr er*

**O'REILLT**

*Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo*

# Inhalt

\	<i>Vorwort von Jeffrey Friedl zur deutschen Übersetzung</i>	<i>xv</i>
i	<i>Vorwort</i>	<i>xvii</i>
	<i>1: Einführung in reguläre Ausdrücke</i>	<i>1</i>
	Probleme aus der Praxis lösen . . . . .	2
	Reguläre Ausdrücke als Programmiersprache . . . . .	3
	Die Analogie zu Dateinamen . . . . .	4
	Die Analogie zu natürlichen Sprachen . . . . .	4
	Reguläre Ausdrücke als Denkweise . . . . .	5
	Textdateien durchsuchen: Egrep . . . . .	6
	Metazeichen bei Egrep . . . . .	7
	Zeilenanfang und Zeilenende . . . . .	8
	Zeichenklassen . . . . .	8
	Auf irgendein Zeichen prüfen: der Punkt . . . . .	11
	Alternation . . . . .	11
	Wörtgrenzen . . . . .	13
	Kurze Rekapitulation . . . . .	15
	Optionale Elemente . . . . .	16
	Andere Quantifier: Repetition . . . . .	16
	Groß- und Kleinschreibung ignorieren . . . . .	18
	Klammern und Rückwärtsreferenzen . . . . .	19
	Ausbrecher! . . . . .	21
	Erweiterung der Fundamente . . . . .	21
	Linguistisches Divertissement . . . . .	21
	Das Ziel eines regulären Ausdrucks . . . . .	22
	Weitere Beispiele . . . . .	22
	Reguläre Ausdrücke: Terminologie . . . . .	24
	Den Status Quo verbessern . . . . .	27

Zusammenfassung . . . . .	29
Persönliche Einsprengsel . . . . .	30
<b>2:            <i>Erweiterte            einführende            Beispiele</i></b>	<b>33</b>
Zu den Beispielen . . . . .	34
Eine kleine Einführung in Perl . . . . .	35
Mustererkennung mit regulären Ausdrücken . . . . .	36
Mehr Praxisnähe . . . . .	38
Seiteneffekte bei erfolgreicher Mustererkennung . . . . .	38
Verschachtelte reguläre Ausdrücke . . . . .	42
Rekapitulation . . . . .	45
Mit regulären Ausdrücken Text verändern . . . . .	47
Automatisiertes Editieren von Dateien . . . . .	49
Ein kleines Mail-Programm . . . . .	50
Verdoppelte Wörter, nochmals. . . . .	57
<b>3: <i>Features und Geschmacksrichtungen</i></b>	<b>63</b>
Ein Spaziergang durch die Landschaft der regulären Ausdrücke. . . . .	64
Grep, und wie es die Welt sah . . . . .	64
The Times They Are a-Changin' . . . . .	66
Kurzer Überblick . . . . .	67
POSIX . . . . .	68
Wartung und Pflege von regulären Ausdrücken . . . . .	71
Eine Regex identifizieren . . . . .	71
Eine erkannte Regex anwenden . . . . .	71
Reguläre Ausdrücke in anderen Programmen . . . . .	72
Wartung und Pflege: Zusammenfassung . . . . .	75
Motoren und Zierleisten. . . . .	75
Chrom und Styling . . . . .	75
Motoren und Fahrer. . . . .	76
Übliche Metazeichen . . . . .	76
»Abkürzungs«-Metazeichen . . . . .	76
Strings als reguläre Ausdrücke. . . . .	80
Zeichenklassen-Kürzel, Punkt und Zeichenklassen . . . . .	82
Anker. . . . .	86
Gruppieren und Einfangen. . . . .	87
Quantifier. . . . .	88
Alternation . . . . .	89
Führer durch die Kapitel für Fortgeschrittene. . . . .	89
Programmspezifische Informationen . . . . .	90

<b>4: Wie Regex-Maschinen arbeiten</b>	<b>91</b>
Motor anlassen!	91
Zwei Arten von Motoren	91
Kalifornische Abgasvorschriften	92
Typen von Regex-Maschinen	92
Aus der Abteilung für Redundanz-Abteilung	94
Grundlegendes zum Matching	95
Zu den Beispielen	95
Regel 1: Der früheste Treffer gewinnt	95
Das »Getriebe« schaltet zum nächsten Zeichen	96
Bestandteile der Regex-Maschine	97
Regel 2: Manche Metazeichen sind gierig	98
Regex-gesteuerte und textgesteuerte Maschinen	102
NFA-Maschine: Regex-gesteuert	102
DFA-Maschine: Textgesteuert	103
Die Geheimnisse des Lebens werden aufgedeckt	105
Backtracking	106
Eine bröcklige Analogie	106
Zwei wichtige Punkte zum Backtracking	107
Gespeicherte Zustände	108
Backtracking und Gier	110
Mehr Gieriges	112
Probleme gierigen Verhaltens	112
Mehrbuchstabile »Anführungszeichen«	113
Genügsamkeit?	114
Gieriges Verhalten bevorzugt immer ein Matching	115
Ist die Alternation gierig?	116
Verwendung von nicht-gieriger Alternation	117
Gierige Alternation im größeren Zusammenhang	118
Zeichenklassen und Alternation	119
NFA, DFA und POSIX	120
Der »längste früheste Treffer«	120
POSIX und der »längste früheste Treffer«	121
Geschwindigkeit und Effizienz	122
DFA und NFA im Vergleich	123
Regex-Methoden aus der Praxis	125
Mitwirkende Faktoren	126
Genau formulieren	126
Schwieriges und Unmögliches	130

Ungewollte Matchings vermeiden . . . . .	131
Eingefassten Text erkennen . . . . .	133
Erwartete Daten und Annahmen . . . . .	136
Weitere gierige Beispiele . . . . .	136
Zusammenfassung . . . . .	140
Zusammenfassung - Mechanismus eines Matchings . . . . .	140
Praktische Auswirkungen des Mechanismus eines Matchings . . . . .	141

**5: Reguläre Ausdrücke gekonnt schreiben** **143**

Ein ernüchterndes Beispiel . . . . .	144
Eine einfache Änderung — Schokoladenseite zuerst . . . . .	145
Gieriges Verhalten nur lokal zulassen . . . . .	145
Zurück zur Realität . . . . .	148
Backtracking global betrachtet . . . . .	149
Überstunden für den POSIX-NFA . . . . .	151
Mehr Arbeit bei einem Fehlschlag . . . . .	151
Einschränkendere Formulierung . . . . .	151
Alternationen können teuer sein . . . . .	152
Ein starker Auftritt . . . . .	153
Auswirkung von Klammern . . . . .	154
Interne Optimierungen . . . . .	158
»Erstes Zeichen«-Optimierung . . . . .	158
Test auf simple Strings . . . . .	159
Einfache Repetition . . . . .	159
Unnötige kleine Quantifier . . . . .	160
Längenerkennung . . . . .	161
Frühes Erkennen des längsten Treffers . . . . .	161
Erkennen, was eigentlich gefragt ist . . . . .	161
String- und Zeilenanker . . . . .	162
Kompilations-Caching . . . . .	162
Maschinentyp ermitteln . . . . .	164
DFA oder NFA? . . . . .	164
Traditioneller NFA oder POSIX-NFA? . . . . .	165
Die Schleife aufbrechen . . . . .	166
Methode 1: Eine Regex aus früheren Erfahrungen aufbauen . . . . .	166
Ein Rezept zum Aufbrechen von Schleifen . . . . .	168
Methode 2: Die kritische Schleife im größeren Zusammenhang . . . . .	170
Methode 3: Ein Internet-Hostname in Anführungszeichen . . . . .	171
Beobachtungen . . . . .	172

C-Kommentare aufbrechen . . . . .	172
Augenschmerzen . . . . .	173
Ein naiver Ansatz . . . . .	173
Die C-Schleife aufbrechen . . . . .	175
Die frei fließende Regex . . . . .	177
Eine helfende Hand führt die Maschine . . . . .	177
Eine gut geführte Regex ist eine schnelle Regex . . . . .	178
Zusammenfassung . . . . .	180
Denken! . . . . .	181
Irrungen und Wirrungen bei Optimierungen . . . . .	181
<b>6: Programmspezifische Informationen</b> . . . . .	<b>185</b>
Fragen, die man stellen sollte . . . . .	185
Etwas so Einfaches wie Grep . . . . .	186
In diesem Kapitel . . . . .	187
Awk . . . . .	188
Unterschiede zwischen Awk-Varianten . . . . .	188
Regex-Funktionen und -Operatoren bei Awk . . . . .	191
Tel . . . . .	192
Operanden bei Tcls Regex . . . . .	192
Reguläre Ausdrücke in Tel . . . . .	193
Regex-Optimierungen bei Tel . . . . .	195
GNU Emacs . . . . .	195
Emacs Strings als reguläre Ausdrücke . . . . .	197
Geschmacksrichtung von Emacs' Regex . . . . .	197
Resultate eines Matchings bei Emacs . . . . .	199
Benchmarking in Emacs . . . . .	201
Optimierungen bei Emacs . . . . .	202
<b>7: Reguläre Ausdrücke in Perl</b> . . . . .	<b>203</b>
The Perl Way . . . . .	205
Reguläre Ausdrücke als Teil der Programmiersprache . . . . .	205
Perls größte Stärken . . . . .	206
Perls größte Schwächen . . . . .	207
Das Huhn oder das Ei, und The Perl Way . . . . .	208
Einführendes Beispiel: CSV Text . . . . .	208
Reguläre Ausdrücke und The Perl Way . . . . .	211
Geschichtliches zu Perl . . . . .	212

Perlges über Regex . . . . .	213
Kontext bei Ausdrücken . . . . .	214
Dynamisches Scoping: Auswirkungen auf die Mustererkennung . . . . .	215
Durch das Matching gesteuerte Spezial-Variablen . . . . .	221
Interpolation von Variablen in Strings . . . . .	223
Perls Regex-Geschmacksrichtung . . . . .	229
Quantifier - gierige und genügsame . . . . .	229
Gruppieren . . . . .	231
String-Anker . . . . .	236
Der Anker für mehrfaches Matching . . . . .	240
Wort-Anker . . . . .	244
Abkürzungszeichen und andere Metazeichen . . . . .	246
Zeichenklassen . . . . .	248
Textmodifikationen »unterwegs«: Lügengeschichten . . . . .	250
Der Match-Operator . . . . .	251
Begrenzungszeichen für den Regex-Operanden . . . . .	251
Modifizier des Match-Operators . . . . .	253
Der Suchstring-Operand . . . . .	255
Andere Seiteneffekte des Match-Operators . . . . .	256
Rückgabewert des Match-Operators . . . . .	256
Einflüsse von außen auf den Match-Operator . . . . .	259
Der Substitutions-Operator . . . . .	260
Der Ersatztext-Operand . . . . .	260
Der /e-Modifizier . . . . .	261
Kontext und Rückgabewert . . . . .	263
Der /g-Modifizier mit einer Regex, die auf »gar nichts« passen kann . . . . .	264
Der Split-Operator . . . . .	264
Grundlegendes zu split . . . . .	264
Komplexes Splitting . . . . .	266
Der Match-Operand bei komplexen Formen von split . . . . .	267
split im skalaren Kontext . . . . .	269
Der Match-Operand mit einfangenden Klammern . . . . .	269
Effizienz in Perl . . . . .	270
»Viele Wege führen zum Ziel« . . . . .	271
Effizienz bei der Kompilation und der /o-Modifizier . . . . .	273
Das asoziale \$& und seine Kumpane . . . . .	279
Leistungseinbuße durch den /i-Modifizier . . . . .	283
Effizienz bei der Substitution . . . . .	287
Benchmarks . . . . .	290

Debug-Informationen zu regulären Ausdrücken . . . . .	291
Die »study«-Funktion . . . . .	293
Das Puzzle zusammensetzen . . . . .	295
Leerzeichen am Anfang und am Ende entfernen. . . . .	296
Große Zahlen in Dreiergruppen aufteilen. . . . .	297
Kommentare aus C-Programmen entfernen. . . . .	298
Prüfen einer E-Mail-Adresse. . . . .	300
Abschließende Bemerkungen. . . . .	311
Anmerkungen zu Perl4. . . . .	312
<i>A: Online-Informationen</i>	<i>317</i>
<i>B: Die E-Mail-Regex</i>	<i>321</i>
<i>Index</i>	<i>325</i>



# Tabellen

1-1	Zusammenfassung der bisher aufgetretenen Metazeichen . . . . .	.15
1-2	Zusammenfassung der Quantifier »Repetitions-Metazeichen« . . . . .	.18
1-3	Zusammenfassung: Metazeichen bei Egrep . . . . .	.29
3-1	Summarischer Überblick über die Geschmacksrichtungen einiger Programme	67
3-2	POSIX Regex-Varianten. . . . .	.69
3-3	Einige Programme und unterstützte Abkürzungen für Metazeichen. . . . .	.77
3-4	String- und Zeilenanker, Zusammenspiel mit Newlines. . . . .	.86
4-1	Einige Programme und ihre Regex-Maschinentypen. . . . .	.94
5-1	Effizienz bei einem traditionellen NFA. . . . .	.147
5-2	Schleife aufbrechen: Beispiele. . . . .	.167
5-3	Schleife aufbrechen: Komponenten für C-Kommentare. . . . .	.176
6-1	Geschmacksrichtungen von einigen häufig benutzten Programmen. . . . .	.186
6-2	Einige grep-Programme unter der Lupe. . . . .	.187
6-3	Einige Awk-Implementierungen genauer betrachtet. . . . .	.189
6-4	Geschmacksrichtung des NFA von Tel. . . . .	.193
6-5	Grundlegende Suchfunktionen von GNU Emacs. . . . .	.196
6-6	String-Metazeichen bei GNU Emacs. . . . .	.197
6-7	Geschmacksrichtung von Emacs' NFA. . . . .	.198
6-8	Emacs' Syntaxklassen. . . . .	.199
7-1	Elemente von regulären Ausdrücken in Perl, Überblick. . . . .	.206
7-2	Elemente von Perls Regex-Sprache, Überblick. . . . .	.207
7-3	Die Bedeutung von local. . . . .	.216
7-4	Quantifier in Perl (gierige und genügsame). . . . .	.229
7-5	Suchmodi im Zusammenhang mit Newlines. . . . .	.237
7-6	String-/Zeilenanker und Punkt in verschiedenen Modi. . . . .	.240
7-7	Regex-Abkürzungszeichen und andere Metazeichen. . . . .	.246

*Tabellen*

7-8	Textmodifikationen »unterwegs« bei Strings und Regex-Operanden. . . . .	250
7-9	Beispiele zu m/-/g und Regex, die leere Treffer zulassen. . . . .	254
7-10	Unartige Module in der Standardbibliothek (die \$_ usw. benutzen). . . . .	283
7-11	Eine einigermaßen formale Syntax für E-Mail-Adressen. . . . .	301