

Vergleich Shell-Globbing mit Regulären Ausdrücken

(C) 2008-2016 T.Birnthaler/H.Gottschalk <howtos(at)ostc.de>
OSTC Open Source Training and Consulting GmbH
<http://www.ostc.de>

\$Id: shell-glob-regex-cmp-HOWTO.txt,v 1.13 2019/11/26 19:37:07 tsbirn Exp \$

Dieses Dokument beschreibt die Unterschiede und Gemeinsamkeiten des "Shell-Globbing" (d.h. den Shell-Mustern zur Dateinamen-Expansion) und der "Regulären Ausdrücke" (d.h. den Mustern zur Texterkennung und Ersetzung in Programmen).

INHALTSVERZEICHNIS

- 1) Einführung
- 2) Vergleichstabelle
- 3) Besonderheiten
- 4) Beispiele

1) Einführung

Obwohl die Sonderzeichen beim Shell-Globbing und in Regulären Ausdrücken teilweise identisch sind, gibt es doch eine Reihe von Unterschieden, die leicht zu Verwechslungen führen.

Shell-Globbing ist "schwächer" in seinen Ausdrucksmöglichkeiten als Reguläre Ausdrücke. Der größte Unterschied ist die "automatische Verankerung" bei den Shell-Mustern (Std: Dateiname muss VOLLSTÄNDIG zum Suchmuster passen), die bei den Regulären Ausdrücken durch "^" und "\$" erzwungen werden muss (Std: nur TEIL eines Textes muss zum Suchmuster passen).

Die Unterschiede zwischen den beiden Verfahren existieren aus pragmatischen Gründen:

- * Shell-Globbing bezieht sich auf Dateinamen
--> Arbeitsgeschwindigkeit auf der Shell-Kommandozeile soll hoch sein, d.h. häufig benötigte Aufgaben sollen einfach + schnell einzutippen sein.
- * Reguläre Ausdrücke beziehen sich auf Textzeilen/Zeichenketten/Datensätze
--> Sollen sehr leistungsfähig sein, daher dürfen sie komplizierter + langsamer einzutippen sein.

2) Vergleichstabelle

	Shell ("glob")	RegExp /.../
Verankerung GROSS/kleinschreibung GROSS/kl. ignorieren	Automatisch Betriebssystemabhängig Betriebssystemabhängig	Nicht automatisch Automatisch /.../i oder -i
Quotierung Zeilenanfang (Anker) Zeilenende (Anker) 1 belieb. Zeichen Bel. viele bel. Z. Analog ALLE Zeichen	\ "..." Automatisch Automatisch ? (außer "/") * (außer "/") ** (inkl. "/")	\ (NUR!) ^ \$. (außer "\n") .* (außer "\n") [\s\S]* (inkl. "\n")
Zeichenwiederh. 0-N Zeichenwiederh. 1-N Zeichenwiederh. M-N Optional (0-1) Ein Element in Liste Alle außer einem Elem.	* (... ...) + (... ...) Nicht möglich ? (... ...) @ (... ...) !(... ...)	x* x+ x{M,N} x? Nicht möglich Nicht möglich
Zeichenklasse Zeichenbereich Neg. Zeichenklasse	[abc] [a-z] [!abc] [^abc]	[abc] [a-z] [^abc]
ZKl. wiederh. (0-N) ZKl. wiederh. (1-N) ZKl. wiederh. (M-N) ZKl. optional (0-1)	Nicht möglich Nicht möglich Nicht möglich Nicht möglich	[abc]* [abc]+ [abc]{M,N} [abc]?
Alternative/Oder Klammern	{M1,M2,...} Nicht möglich	M1 M2 ... (...)

Metazeichen als normale Zeichen behandeln:

Stern	*	*
Fragezeichen	\?	\?
Punkt	\.	\.
Caret	\^	\^
Dollarzeichen	\\$	\\$
Leerzeichen	\ oder " " oder ' '	" " oder []
Eckige Klammern	\[\]	\[\]

Vollständiger Match notwendig bei Shell automatisch, bei Regex manuell:

Keine Verankerung	*abc*	abc
Links verankert	abc*	^abc
Rechts verankert	*abc	abc\$
Voll verankert	abc	^abc\$

3) Besonderheiten

Das LEERZEICHEN ist beim Shell-Globbing ein Sonderzeichen, um mehrere Muster zu trennen. Soll es Teil des Dateinamens sein, muss es mit "\" oder "...\" oder '...\" geschützt werden.

In Regulären Ausdrücken ist das Leerzeichen ein ganz normales Zeichen (außer ^_er im Extended-Modus /.../x).

Der "." zu Beginn von Dateinamen wird beim Shell-Globbing besonders behandelt, indem er von "?" und "*" nicht gefunden wird (versteckte Dateien).

Das Zeichen "/" (Verzeichnistrenner) wird beim Shell-Globbing von "*" und "?" nicht gematcht (nur von "***").

Das Zeichen "\n" (Zeilenvorschub) wird bei Regulären Ausdrücken von "." nicht gematcht (durch Tricks wie "\n", "[\s\S]" oder Option "s" (single line) in Perl aber dennoch möglich).

4) Beispiele

* Beispiele für Regex-Muster und ihre Treffer (*) bzw. Nicht-Treffer (-) (Strings, Zeilen)

	/a/	./	/[abc]/	/[a-z]/	/^[a-z]/	/^[a-z]/	/[a-z]\$/	/^[a-z]\$/
d	-	*	-	*	-	*	*	*
made	*	*	*	*	-	*	*	-
abc	*	*	*	*	-	*	*	-
aaaa	*	*	*	*	-	*	*	-
xyz	-	*	-	*	-	*	*	-
" "	-	-	-	-	-	-	-	-
" "	-	*	-	-	*	-	-	-
" "	-	*	-	-	*	-	-	-
*	-	*	-	-	*	-	-	-

* Beispiel für Shell-Muster und ihre Treffer (*) bzw. Nicht-Treffer (-) (Dateinamen, glob)

	a	*?*	*[abc]*	*[a-z]*	*[!a-z]*	[a-z]*	*[a-z]	[a-z]
d	-	*	-	*	-	*	*	*
made	*	*	*	*	-	*	*	-
abc	*	*	*	*	-	*	*	-
aaaa	*	*	*	*	-	*	*	-
xyz	-	*	-	*	-	*	*	-
" "	-	-	-	-	-	-	-	-
" "	-	*	-	-	*	-	-	-
" "	-	*	-	-	*	-	-	-
*	-	*	-	-	*	-	-	-