

LPI Essentials Chapter 7 — Regular Expressions and grep

Alexander Voigt

Technische Universität Dresden
Institut für Kern- und Teilchenphysik

IKTP Computing Seminar
26 September 2012



① Regular Expressions

- Definition

- Meta Symbols

- Priority of Operators

- Examples

② grep

- grep usage

- grep Examples

Definition

A regular expression (RE) describes a set of words of a formal language in the following way:

Let $\Sigma = \{a, b, c, \dots\}$ be a set of symbols (alphabet), then the following constants are REs:

- ϵ , the empty string
- $a, \forall a \in \Sigma$

If r and s are REs, then the following are REs:

- (rs) (concatenation)
- $(r|s)$ (alternation)
- (r^*) (Kleene star)

$\forall a \in \Sigma$ one defines

- $a^+ := aa^*$
- $a? := (a|\epsilon)$

Meta Symbols

meta symbol		description	examples
ERE	BRE		
[]		match single character within []	[abc], [a-z]
[^]		match single character not in []	[^abc], [^a-z]
()	\(\)	marked subexpression	(a b)
{ }	\{ \}	match between m and n times	a{1,3}
	\\	choice operator	a b
?	\\?	match zero or one time	a?
+	\\+	match one or more times	a+
*		match zero or more times	a*
-		define range	[a-zA-Z0-9]
^		match begin of line	^abc
\$		match end of line	xyz\$
.		match any single character	ab.de
\\n		nth marked subexpression	(ab)\\1
\\<		match begin of word	\\<word\\>
\\>		match end of word	\\<word\\>

Priority of Operators

Meta symbols in descending priority: *, concatenation, |

Examples:

$(ab)c = abc$

$a|(b(c*)) = a|bc*$

Examples

RE	matches	does not match
ab	ab	a, b, c
a(bc)*	a, abc, abcabc	ab
abc*	ab, abc, abcc	abcabc
abc+	abc, abcc	ab
a(b c)d	abd, acd	ad, abcd
ab?c	ac, abc	a, ab
(ab)?c	c, abc	a, ab
ab{2,4}	abb, abbb, abbbb	a, ab, abbbbb
(ab)\1	abab	a, ab, abb
a b* = a (b*)	ε, a, b, bb, bbb	aa, aba

Trick: Use `egrep` to test these!

```
1 echo "abc" | egrep -o "a(bc)*"
```

grep usage

grep searches for a regular expression <regex> in files or stdin.

usage:

```
1 grep [options] "<regex>" files
```

or

```
1 echo "foo bar" | grep [options] "<regex>"
```

option	description
-c	count number of matching lines
-i	ignore capitalization
-l	list only file names
-n	print line and line number
-r	search recursively also in subfolders
-v	print only lines, that do not match
-o	print only the matched expression
-E	use ERE, instead of BRE (BRE is the default)

Note: Use `egrep` instead of `grep -E`

grep Examples

```
1 $ grep MSSM paper.tex
2 familiar CMSSM, the \cESSM\ has no $\mu$ and $B\mu$
3 be adjusted to fulfill SM constraints. Hence, the CMSSM
4 intermediate matching to the MSSM because the MSSM and the
5 MSSM (CMSSM). Typically these employ two-loop
```

```
1 $ grep -n ^MSSM paper.tex
2 856:MSSM (CMSSM). Typically these employ two-loop
```

```
1 $ grep "\<MSSM\>" paper.tex
2 intermediate matching to the MSSM because the MSSM and the
3 MSSM (CMSSM). Typically these employ two-loop
```

```
1 $ grep -c "\<MSSM\>" paper.tex
2 2
```