CSCI 330
The UNIX System

Regular Expressions

Regular Expression

- A pattern of special characters used to match strings in a search
- Typically made up from special characters called metacharacters

- Regular expressions are used thoughout UNIX:
  - Editors: ed, ex, vi
  - Utilities: grep, egrep, sed, and awk

Metacharacters

RE Metacharacter	Matches…
.	**Any one character, except new line**
[a-z]	**Any one of the enclosed characters (e.g. a-z)**
*****	**Zero or more of preceding character**
? or \?	**Zero or one of the preceding characters**
+ or \+	**One or more of the preceding characters**

- any non-metacharacter matches itself

The grep Utility

- “grep” command:

  searches for text in file(s)

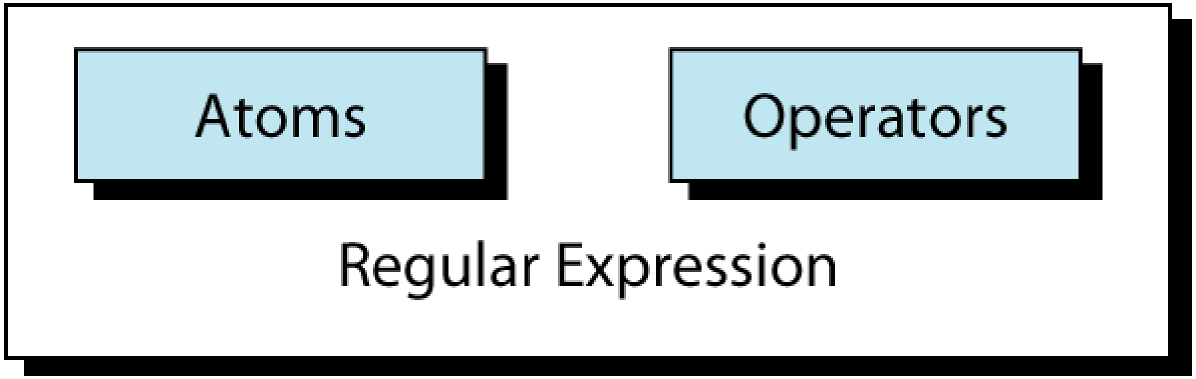
<u>Examples:</u>

```
% grep root mail.log
% grep r..t mail.log
% grep ro*t mail.log
% grep ‘ro*t’ mail.log
% grep ‘r[a-z]*t’ mail.log
```

MORE METACHARACTERS

RE Metacharacter	Matches…
^	**beginning of line**
$	**end of line**
\*char*	**Escape the meaning of *char* following it**
[^]	**One character not in the set**
\\<	**Beginning of word anchor**
\\>	**End of word anchor**
( ) or \\(\\)	**Tags matched characters to be used later (max = 9)**
| or \\\|	**Or grouping**
x\\{m\\}	**Repetition of character x, m times (x,m = integer)**
x\\{m,\\}	**Repetition of character x, at least m times**
x\\{m,n\\}	**Repetition of character x between m and m times**

Regular Expression

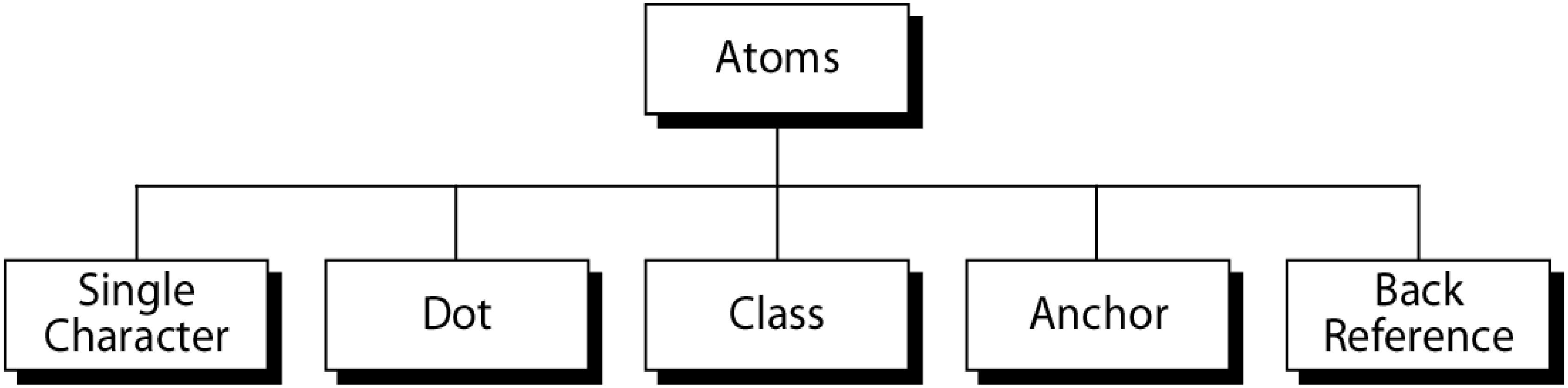


An atom specifies what text is to be matched and where it is to be found.

An operator combines regular expression atoms.

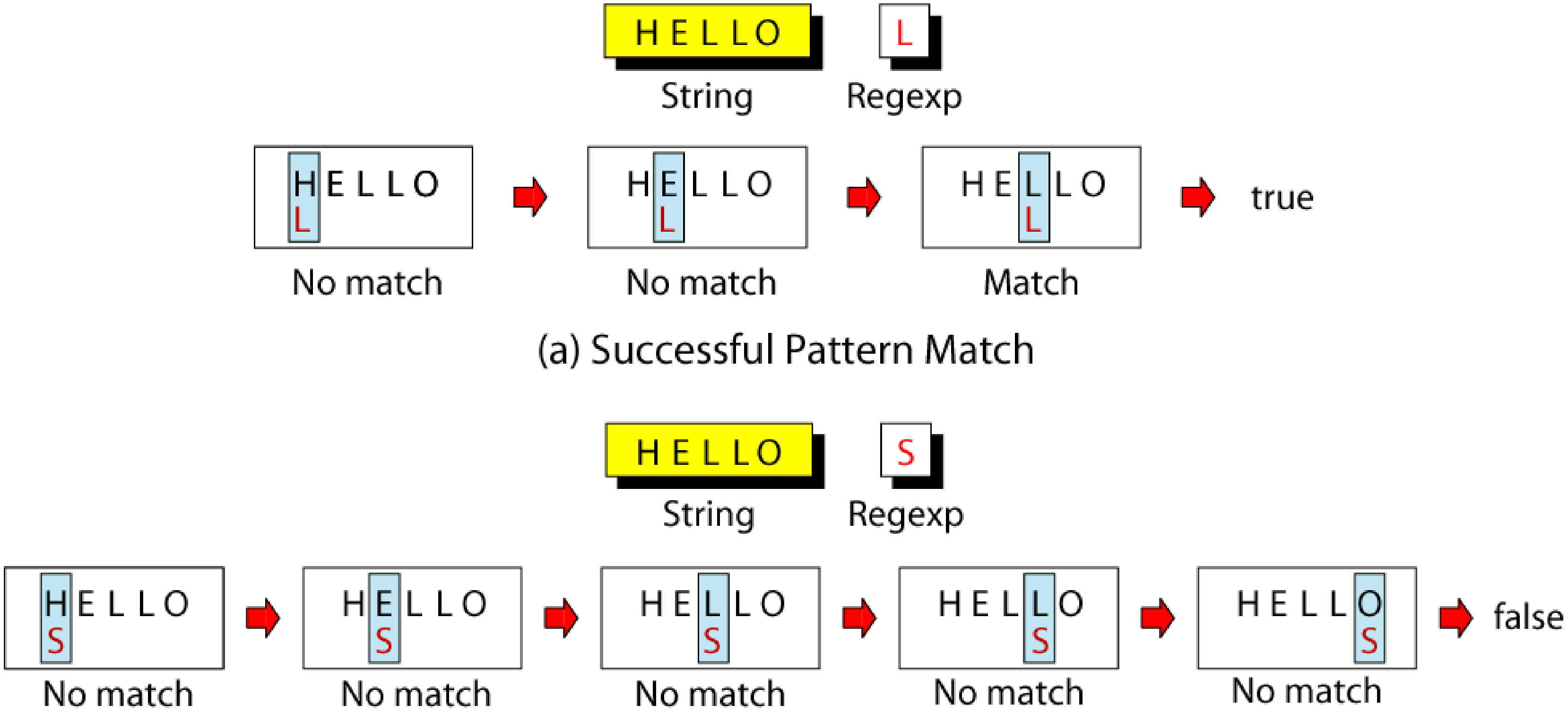
Atoms

An atom specifies what text is to be matched and where it is to be found.



Single-Character Atom

A single character matches itself

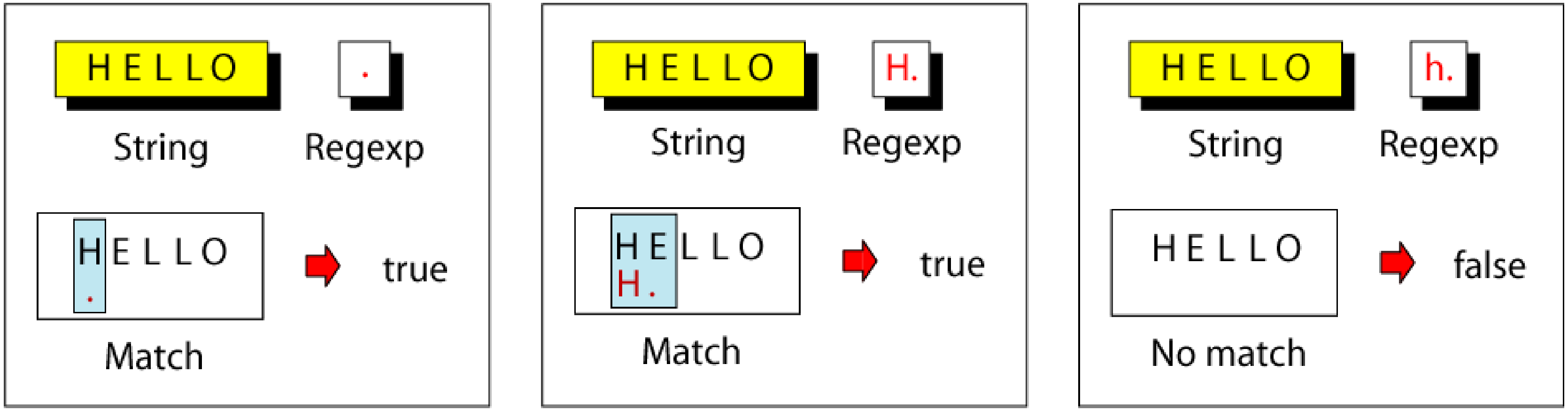


(a) Successful Pattern Match

(b) Unsuccessful Pattern Match

Dot Atom

matches any single character except for a new line character (\n)

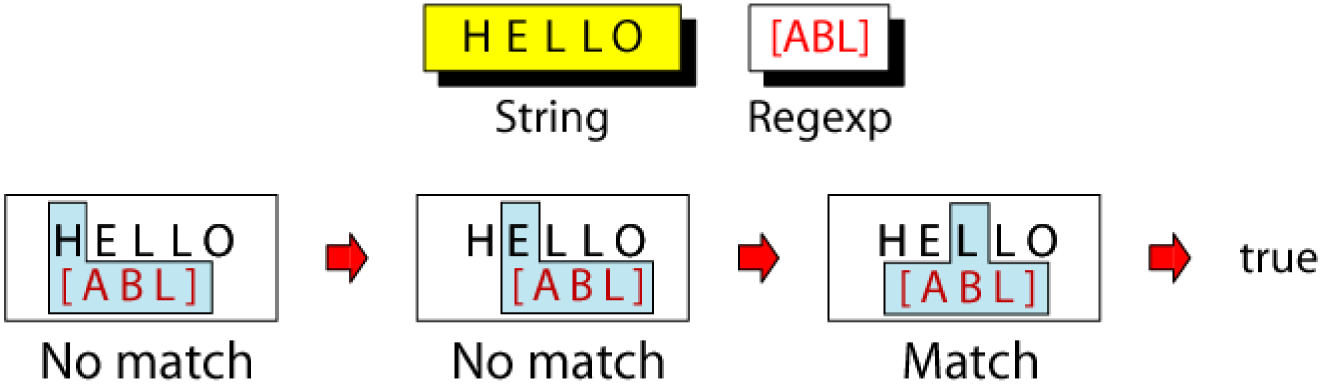


(a) Single-Character (b) Combination–True (c) Combination–False

Class Atom

matches only single character that can be any of the characters defined in a set:

Example: [ABC] matches either A, B, or C.



Notes:

1) A range of characters is indicated by a dash, e.g. [A-Q]
2) Can specify characters to be excluded from the set, e.g. [^0-9] matches any character other than a number.

Example: Classes

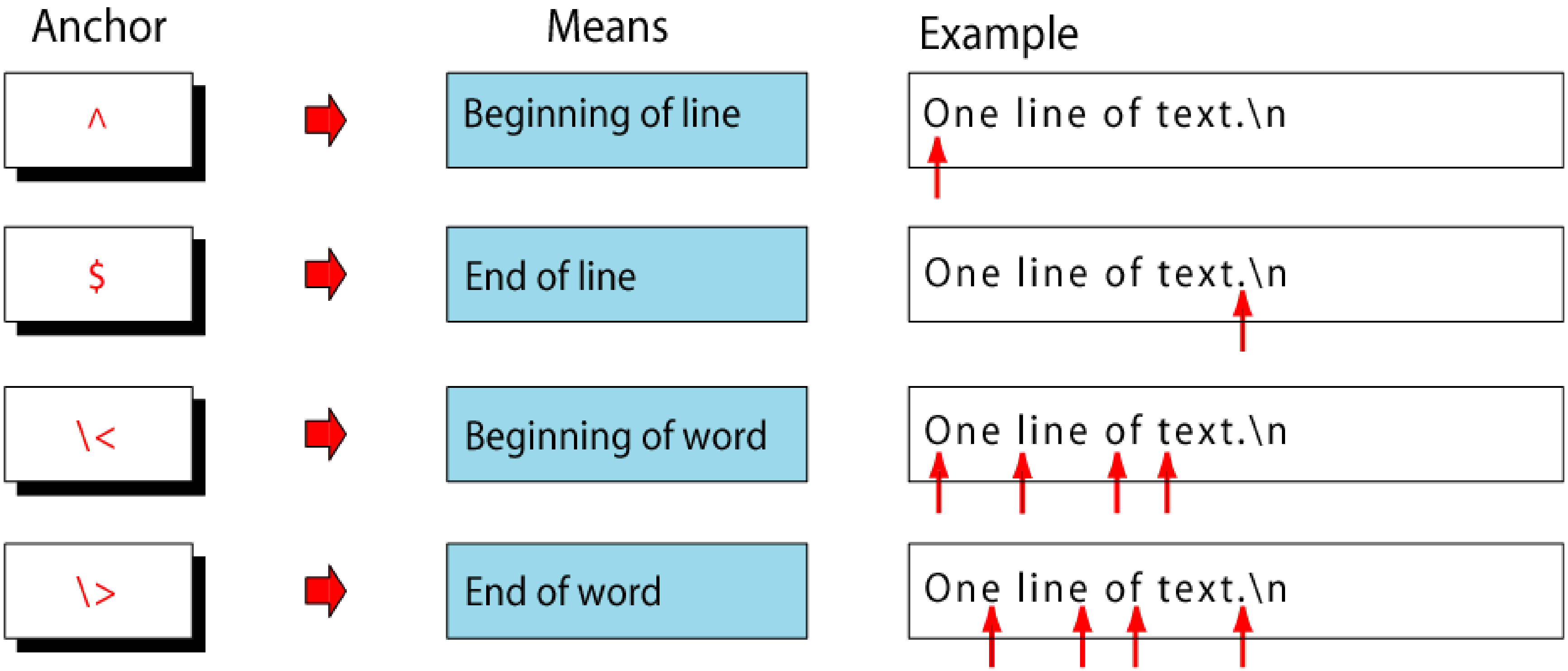
RegExpr	Means	RegExpr	Means
[A-H]	[ABCDEFGH]	[^AB]	Any character except A or B
[A-Z]	Any uppercase alphabetic	[A-Za-z]	Any alphabetic
[0-9]	Any digit	[^0-9]	Any character except a digit
[[a]	[or a	[]a]	] or a
[0-9\-]	digit or hyphen	[^\^]	Anything except^

SHORT-HAND CLASSES

- [:alnum:]
- [:alpha:]
- [:upper:]
- [:lower:]
- [:digit:]
- [:space:]

Anchors

Anchors tell where the next character in the pattern must be located in the text data.



Anchor	Means	Example
^	Beginning of line	One line of text.\n
$	End of line	One line of text.\n
\<	Beginning of word	One line of text.\n
\>	End of word	One line of text.\n

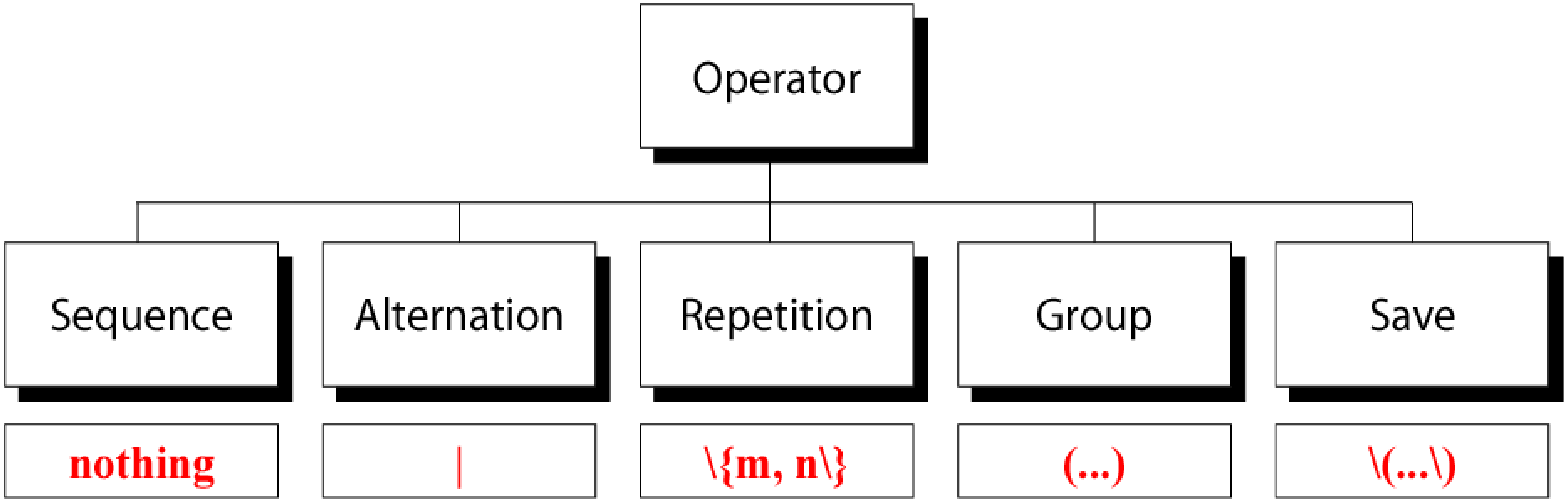
Back References: \n

- used to retrieve saved text in one of nine buffers
- can refer to the text in a saved buffer by using a back reference:

  ex.: \1 \2 \3 ...\9

- more details on this later

Operators



Operator

Sequence

Alternation

Repetition

Group

Save

nothing

|

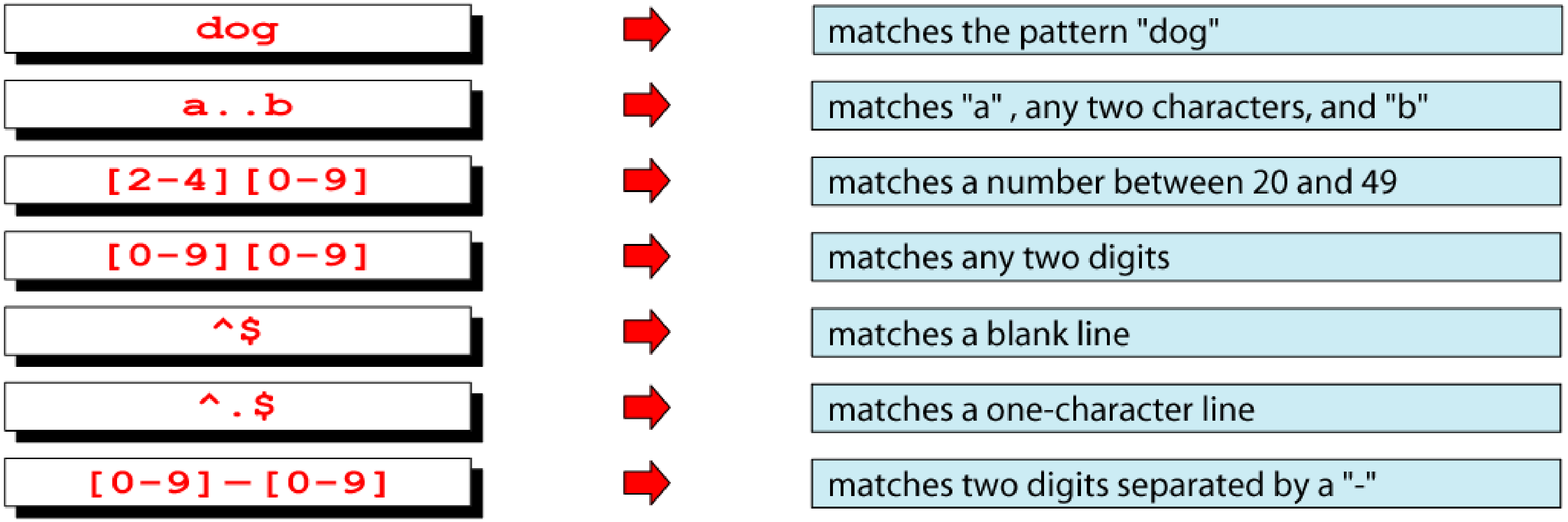
\{m, n\}

(...)

\(...\)

Sequence Operator

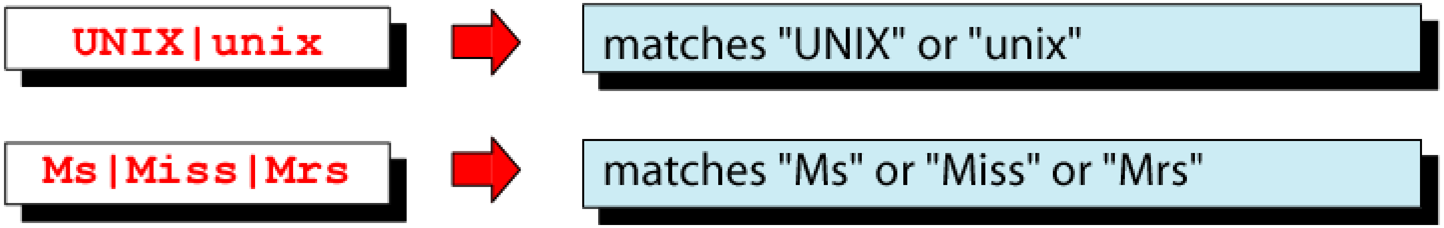
In a sequence operator, if a series of atoms are shown in a regular expression, there is no operator between them.



`dog`	matches the pattern "dog"
`a..b`	matches "a" , any two characters, and "b"
`[2-4][0-9]`	matches a number between 20 and 49
`[0-9][0-9]`	matches any two digits
`^$`	matches a blank line
`^.$`	matches a one-character line
`[0-9]-[0-9]`	matches two digits separated by a "-"

Alternation Operator: | or \|

operator (| or \|) is used to define one **or** more alternatives



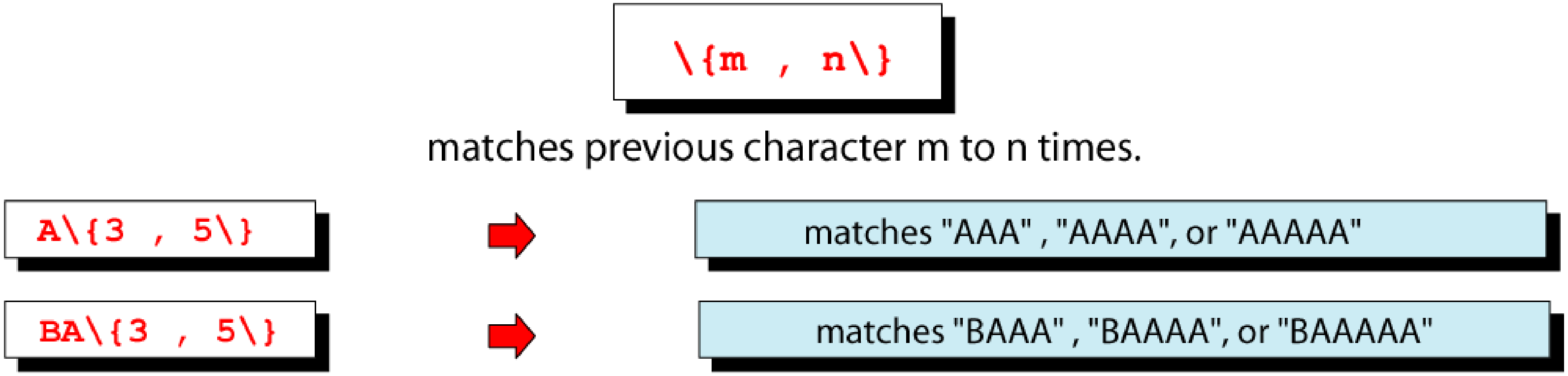
UNIX|unix → matches "UNIX" or "unix"

Ms|Miss|Mrs → matches "Ms" or "Miss" or "Mrs"

Note: depends on version of “grep”

Repetition Operator: \{…\}

The repetition operator specifies that the atom or expression immediately before the repetition may be repeated.



\{m , n\}

matches previous character m to n times.

A\{3 , 5\} → matches "AAA" , "AAAA", or "AAAAA"

BA\{3 , 5\} → matches "BAAA" , "BAAAA", or "BAAAAA"

Basic Repetition Forms

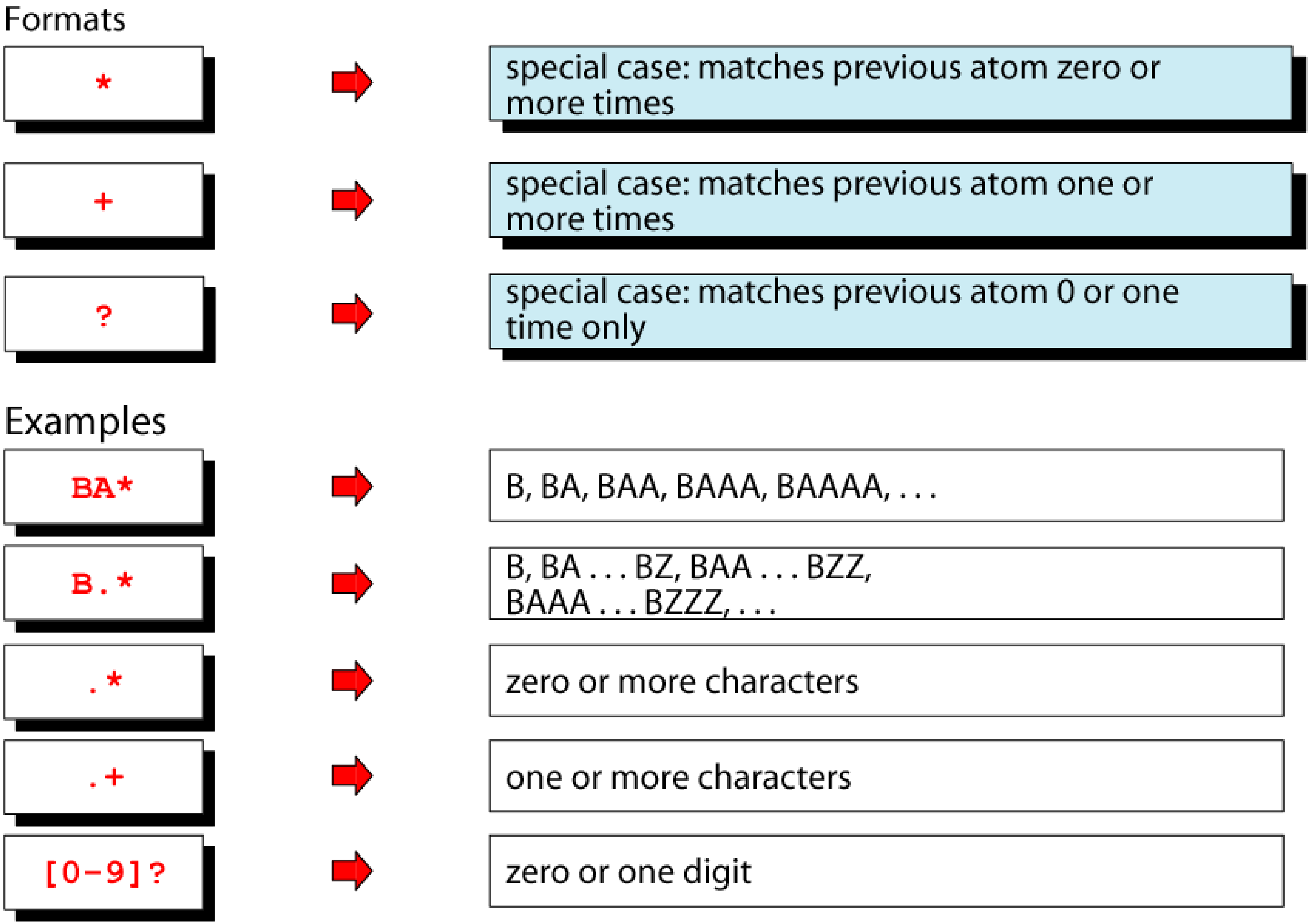
Formats

`\{m\}`	matches previous atom exactly m times
`\{m, \}`	matches previous atom m times or more
`\{, n\}`	matches previous atom n times or less

Examples

`CA\{5\}`	`CAAAAA`
`CA\{3,\}`	`CAAA, CAAAA, CAAAAA, …`
`CA\{,2\}`	`C, CA, CAA`

Short Form Repetition Operators:



Formats

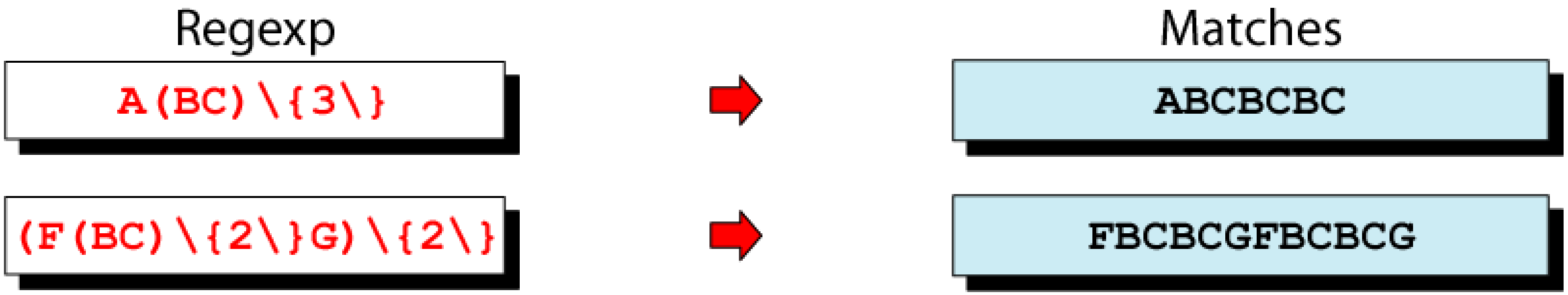
*	special case: matches previous atom zero or more times
+	special case: matches previous atom one or more times
?	special case: matches previous atom 0 or one time only

Examples

BA*	B, BA, BAA, BAAA, BAAAA, . . .
B.*	B, BA . . . BZ, BAA . . . BZZ, BAAA . . . BZZZ, . . .
.*	zero or more characters
.+	one or more characters
[0-9]?	zero or one digit

Group Operator

In the group operator, when a group of characters is enclosed in parentheses, the next operator applies to the whole group, not only the previous characters.



Regexp		Matches
A(BC)\{3\}	➡	ABCBCBC
(F(BC)\{2\}G)\{2\}	➡	FBCBCGFBCBCG

Note: depends on version of "grep"
use \(and \) instead

GREP DETAIL AND EXAMPLES

- grep is family of commands
  - grep

    common version
  - egrep

    understands extended REs

    (| + ? ( ) don't need backslash)
  - fgrep

    understands only fixed strings, i.e. is faster
  - rgrep

    will traverse sub-directories recursively

Commonly used “grep” options:

-c	Print only a count of matched lines.
-i	Ignore uppercase and lowercase distinctions.
-l	List all files that contain the specified pattern.
-n	Print matched lines and line numbers.
-s	Work silently; display nothing except error messages. Useful for checking the exit status.
-v	Print lines that do not match the pattern.

EXAMPLE: GREP WITH PIPE

Pipe the output of the "ls –l" command to grep and list/select only directory entries.

```
% ls -l | grep '^d'
drwxr-xr-x   2 krush    csci        512 Feb  8 22:12 assignments
drwxr-xr-x   2 krush    csci        512 Feb  5 07:43 feb3
drwxr-xr-x   2 krush    csci        512 Feb  5 14:48 feb5
drwxr-xr-x   2 krush    csci        512 Dec 18 14:29 grades
drwxr-xr-x   2 krush    csci        512 Jan 18 13:41 jan13
drwxr-xr-x   2 krush    csci        512 Jan 18 13:17 jan15
drwxr-xr-x   2 krush    csci        512 Jan 18 13:43 jan20
drwxr-xr-x   2 krush    csci        512 Jan 24 19:37 jan22
drwxr-xr-x   4 krush    csci        512 Jan 30 17:00 jan27
drwxr-xr-x   2 krush    csci        512 Jan 29 15:03 jan29
% ls -l | grep -c '^d'
10
```

Display the number of lines where the pattern was found. This does not mean the number of occurrences of the pattern.

Example: grep with \< \>

```
% cat grep-datafile
northwest       NW      Charles Main            300000.00
western         WE      Sharon Gray             53000.89
southwest       SW      Lewis Dalsass           290000.73
southern        SO      Suan Chin               54500.10
southeast       SE      Patricia Hemenway       400000.00
eastern         EA      TB Savage               440500.45
northeast       NE      AM Main Jr.             57800.10
north           NO      Ann Stephens            455000.50
central         CT      KRush                   575500.70
Extra [A-Z]****[0-9]..$5.00
```

Print the line if it contains the word “north”.

```
% grep '\<north\>' grep-datafile
north           NO      Ann Stephens            455000.50
```

EXAMPLE: GREP WITH A\|B

```
% cat grep-datafile
northwest       NW      Charles Main            300000.00
western         WE      Sharon Gray             53000.89
southwest       SW      Lewis Dalsass           290000.73
southern        SO      Suan Chin               54500.10
southeast       SE      Patricia Hemenway       400000.00
eastern         EA      TB Savage               440500.45
northeast       NE      AM Main Jr.             57800.10
north           NO      Ann Stephens            455000.50
central         CT      KRush                   575500.70
Extra [A-Z]****[0-9]..$5.00
```

Print the lines that contain either the expression “NW” or the expression “EA”

```
% grep 'NW\|EA' grep-datafile
northwest       NW      Charles Main            300000.00
eastern         EA      TB Savage               440500.45
```

Note: egrep works with |

Example: egrep with +

```
% cat grep-datafile
northwest       NW      Charles Main            300000.00
western         WE      Sharon Gray             53000.89
southwest       SW      Lewis Dalsass           290000.73
southern        SO      Suan Chin               54500.10
southeast       SE      Patricia Hemenway       400000.00
eastern         EA      TB Savage               440500.45
northeast       NE      AM Main Jr.             57800.10
north           NO      Ann Stephens            455000.50
central         CT      KRush                   575500.70
Extra [A-Z]****[0-9]..$5.00
```

Print all lines containing one or more 3's.

```
% egrep '3+' grep-datafile
northwest       NW      Charles Main            300000.00
western         WE      Sharon Gray             53000.89
southwest       SW      Lewis Dalsass           290000.73
```

Note: grep works with \+

EXAMPLE: EGREP WITH RE: ?

```
% cat grep-datafile
northwest       NW      Charles Main            300000.00
western         WE      Sharon Gray             53000.89
southwest       SW      Lewis Dalsass           290000.73
southern        SO      Suan Chin               54500.10
southeast       SE      Patricia Hemenway       400000.00
eastern         EA      TB Savage               440500.45
northeast       NE      AM Main Jr.             57800.10
north           NO      Ann Stephens            455000.50
central         CT      KRush                   575500.70
Extra [A-Z]****[0-9]..$5.00
```

Print all lines containing a 2, followed by zero or one period, followed by a number.

```
% egrep '2\.?[0-9]' grep-datafile
southwest       SW      Lewis Dalsass           290000.73
```

Note: grep works with \?

EXAMPLE: EGREP WITH ( )

```
% cat grep-datafile
northwest       NW      Charles Main            300000.00
western         WE      Sharon Gray             53000.89
southwest       SW      Lewis Dalsass           290000.73
southern        SO      Suan Chin               54500.10
southeast       SE      Patricia Hemenway       400000.00
eastern         EA      TB Savage               440500.45
northeast       NE      AM Main Jr.             57800.10
north           NO      Ann Stephens            455000.50
central         CT      KRush                   575500.70
Extra [A-Z]****[0-9]..$5.00
```

Print all lines containing one or more consecutive occurrences of the pattern “no”.

```
% egrep '(no)+' grep-datafile
northwest       NW      Charles Main            300000.00
northeast       NE      AM Main Jr.             57800.10
north           NO      Ann Stephens            455000.50
```

Note: grep works with \(\) \+

EXAMPLE: EGREP WITH (A | B)

```
% cat grep-datafile
northwest       NW      Charles Main            300000.00
western         WE      Sharon Gray             53000.89
southwest       SW      Lewis Dalsass           290000.73
southern        SO      Suan Chin               54500.10
southeast       SE      Patricia Hemenway       400000.00
eastern         EA      TB Savage               440500.45
northeast       NE      AM Main Jr.             57800.10
north           NO      Ann Stephens            455000.50
central         CT      KRush                   575500.70
Extra [A-Z]****[0-9]..$5.00
```

Print all lines containing the uppercase letter “S”, followed by either “h” or “u”.

```
% egrep 'S(h|u)' grep-datafile
western         WE      Sharon Gray             53000.89
southern        SO      Suan Chin               54500.10
```

Note: grep works with \(\) \|

EXAMPLE: FGREP

```
% cat grep-datafile
northwest       NW      Charles Main            300000.00
western         WE      Sharon Gray             53000.89
southwest       SW      Lewis Dalsass           290000.73
southern        SO      Suan Chin               54500.10
southeast       SE      Patricia Hemenway       400000.00
eastern         EA      TB Savage               440500.45
northeast       NE      AM Main Jr.             57800.10
north           NO      Ann Stephens            455000.50
central         CT      KRush                   575500.70
Extra [A-Z]****[0-9]..$5.00
```

Find all lines in the file containing the literal string “[A-Z]****[0-9]..$5.00”. All characters are treated as themselves. There are no special characters.

```
% fgrep '[A-Z]****[0-9]..$5.00' grep-datafile
Extra [A-Z]****[0-9]..$5.00
```

EXAMPLE: GREP WITH ^

```
% cat grep-datafile
northwest       NW      Charles Main            300000.00
western         WE      Sharon Gray             53000.89
southwest       SW      Lewis Dalsass           290000.73
southern        SO      Suan Chin               54500.10
southeast       SE      Patricia Hemenway       400000.00
eastern         EA      TB Savage               440500.45
northeast       NE      AM Main Jr.             57800.10
north           NO      Ann Stephens            455000.50
central         CT      KRush                   575500.70
Extra [A-Z]****[0-9]..$5.00
```

Print all lines beginning with the letter n.

```
% grep '^n' grep-datafile
northwest       NW      Charles Main            300000.00
northeast       NE      AM Main Jr.             57800.10
north           NO      Ann Stephens            455000.50
```

Example: grep with $

```
% cat grep-datafile
northwest       NW      Charles Main            300000.00
western         WE      Sharon Gray             53000.89
southwest       SW      Lewis Dalsass           290000.73
southern        SO      Suan Chin               54500.10
southeast       SE      Patricia Hemenway       400000.00
eastern         EA      TB Savage               440500.45
northeast       NE      AM Main Jr.             57800.10
north           NO      Ann Stephens            455000.50
central         CT      KRush                   575500.70
Extra [A-Z]****[0-9]..$5.00
```

Print all lines ending with a period and exactly two zero numbers.

```
% grep '\.00$' grep-datafile
northwest       NW      Charles Main            300000.00
southeast       SE      Patricia Hemenway       400000.00
Extra [A-Z]****[0-9]..$5.00
```

EXAMPLE: GREP WITH \CHAR

```
% cat grep-datafile
northwest       NW      Charles Main            300000.00
western         WE      Sharon Gray             53000.89
southwest       SW      Lewis Dalsass           290000.73
southern        SO      Suan Chin               54500.10
southeast       SE      Patricia Hemenway       400000.00
eastern         EA      TB Savage               440500.45
northeast       NE      AM Main Jr.             57800.10
north           NO      Ann Stephens            455000.50
central         CT      KRush                   575500.70
Extra [A-Z]****[0-9]..$5.00
```

Print all lines containing the number 5, followed by a literal period and any single character.

```
% grep '5\..' grep-datafile
Extra [A-Z]****[0-9]..$5.00
```

EXAMPLE: GREP WITH [ ]

```
% cat grep-datafile
northwest       NW      Charles Main            300000.00
western         WE      Sharon Gray             53000.89
southwest       SW      Lewis Dalsass           290000.73
southern        SO      Suan Chin               54500.10
southeast       SE      Patricia Hemenway       400000.00
eastern         EA      TB Savage               440500.45
northeast       NE      AM Main Jr.             57800.10
north           NO      Ann Stephens            455000.50
central         CT      KRush                   575500.70
Extra [A-Z]****[0-9]..$5.00
```

Print all lines beginning with either a “w” or an “e”.

```
% grep '^[we]' grep-datafile
western         WE      Sharon Gray             53000.89
eastern         EA      TB Savage               440500.45
```

EXAMPLE: GREP WITH [^]

```
% cat grep-datafile
northwest       NW      Charles Main            300000.00
western         WE      Sharon Gray             53000.89
southwest       SW      Lewis Dalsass           290000.73
southern        SO      Suan Chin               54500.10
southeast       SE      Patricia Hemenway       400000.00
eastern         EA      TB Savage               440500.45
northeast       NE      AM Main Jr.             57800.10
north           NO      Ann Stephens            455000.50
central         CT      KRush                   575500.70
Extra [A-Z]****[0-9]..$5.00
```

Print all lines ending with a period and exactly two non-zero numbers.

```
% grep '\.[^0][^0]$' grep-datafile
western         WE      Sharon Gray             53000.89
southwest       SW      Lewis Dalsass           290000.73
eastern         EA      TB Savage               440500.45
```

Example: grep with x\{m\}

```
% cat grep-datafile
northwest       NW      Charles Main            300000.00
western         WE      Sharon Gray             53000.89
southwest       SW      Lewis Dalsass           290000.73
southern        SO      Suan Chin               54500.10
southeast       SE      Patricia Hemenway       400000.00
eastern         EA      TB Savage               440500.45
northeast       NE      AM Main Jr.             57800.10
north           NO      Ann Stephens            455000.50
central         CT      KRush                   575500.70
Extra [A-Z]****[0-9]..$5.00
```

Print all lines where there are at least six consecutive numbers followed by a period.

```
% grep '[0-9]\{6\}\.' grep-datafile
northwest       NW      Charles Main            300000.00
southwest       SW      Lewis Dalsass           290000.73
southeast       SE      Patricia Hemenway       400000.00
eastern         EA      TB Savage               440500.45
north           NO      Ann Stephens            455000.50
central         CT      KRush                   575500.70
```

Example: grep with \<

```
% cat grep-datafile
northwest       NW      Charles Main            300000.00
western         WE      Sharon Gray             53000.89
southwest       SW      Lewis Dalsass           290000.73
southern        SO      Suan Chin               54500.10
southeast       SE      Patricia Hemenway       400000.00
eastern         EA      TB Savage               440500.45
northeast       NE      AM Main Jr.             57800.10
north           NO      Ann Stephens            455000.50
central         CT      KRush                   575500.70
Extra [A-Z]****[0-9]..$5.00
```

Print all lines containing a word starting with “north”.

```
% grep '\<north' grep-datafile
northwest       NW      Charles Main            300000.00
northeast       NE      AM Main Jr.             57800.10
north           NO      Ann Stephens            455000.50
```

SUMMARY

- regular expressions
- for grep family of commands