Univerzitet Singidunum

Web programming

www.singidunum.ac.rs

Regular Expressions

- A pattern of special characters used to match strings in a search
- Typically made up from special characters called metacharacters
- Widely used in programing:
  - JavaScript form validation
  - Regular expressions are used thoughout UNIX:
    - Editors: ed, ex, vi
    - Utilities: grep, egrep, sed, and awk

Metacharacters

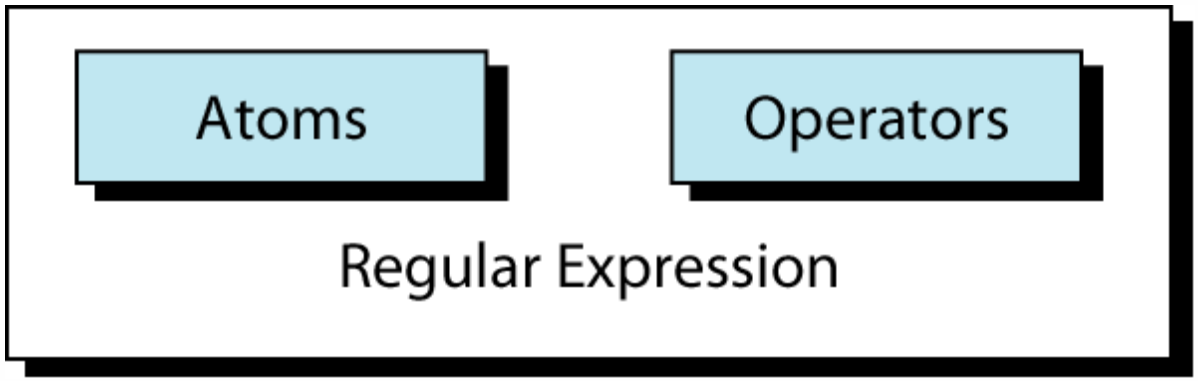
RE Metacharacter	Matches…
.	Any one character, except new line
[a-z]	Any one of the enclosed characters (e.g. a-z)
*	Zero or more of preceding character
? or \?	Zero or one of the preceding characters
+ or \+	One or more of the preceding characters

- any non-metacharacter matches itself

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

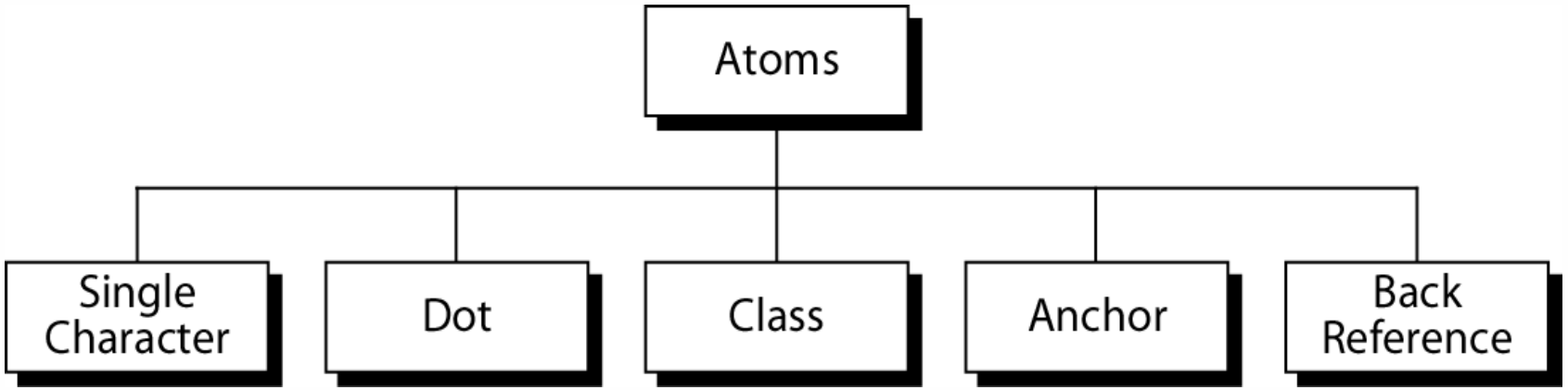


An atom specifies what text is to be matched and where it is to be found.

An operator combines regular expression atoms.

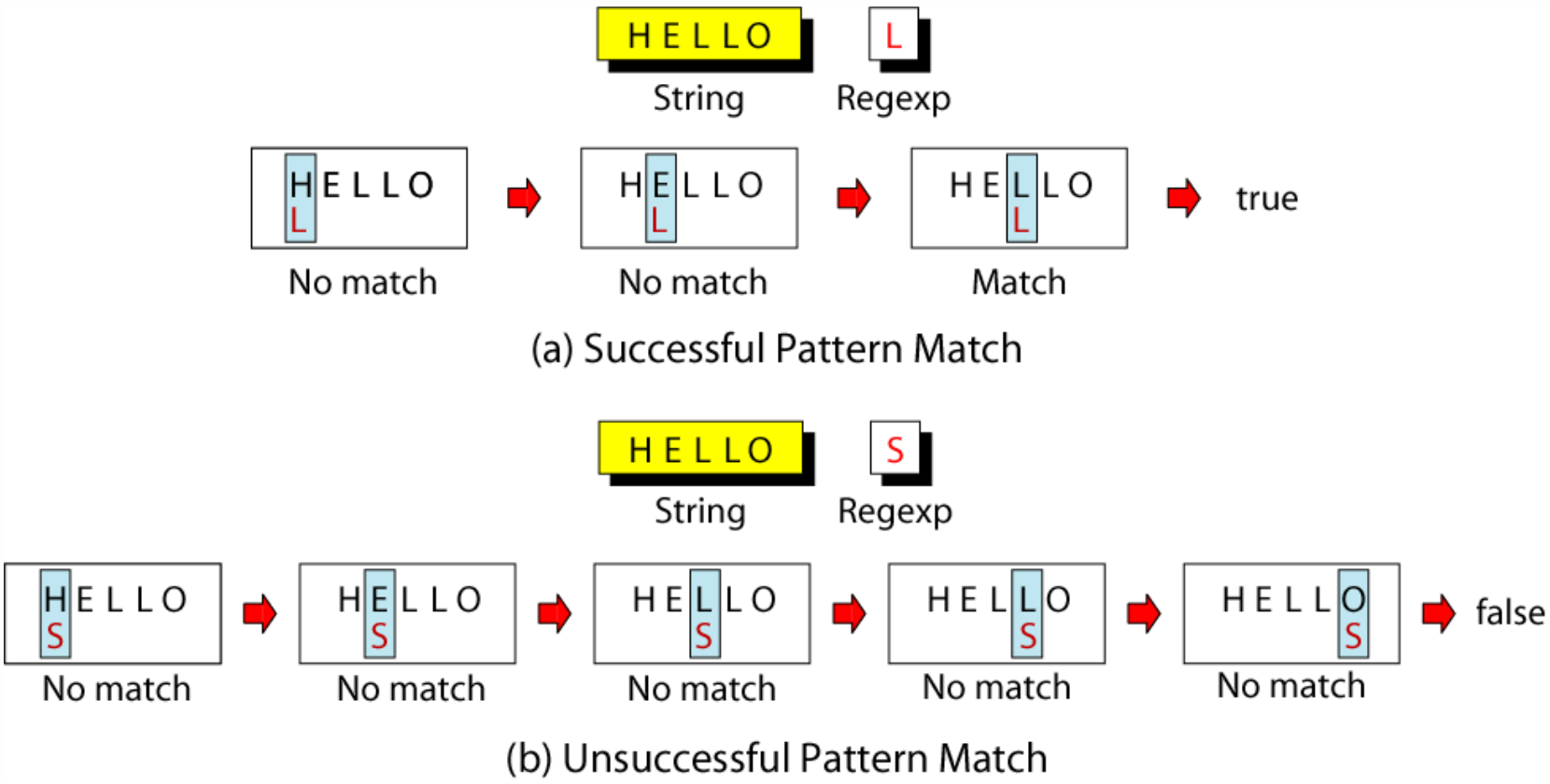
Atoms

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



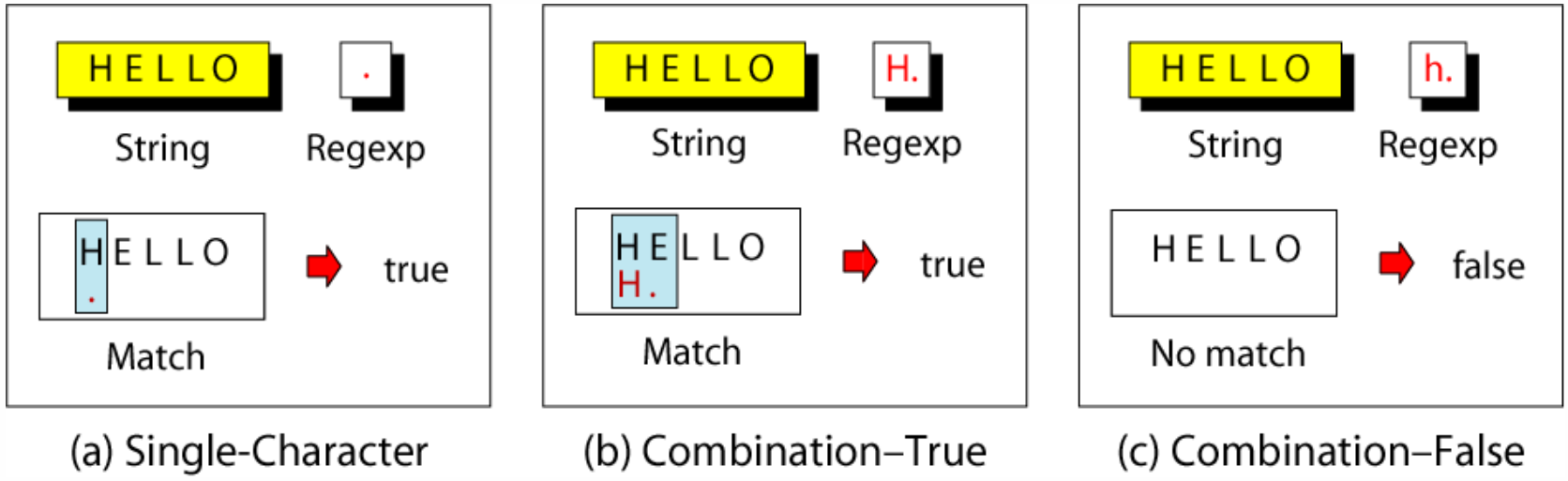
Single-Character Atom

A single character matches itself



Dot Atom

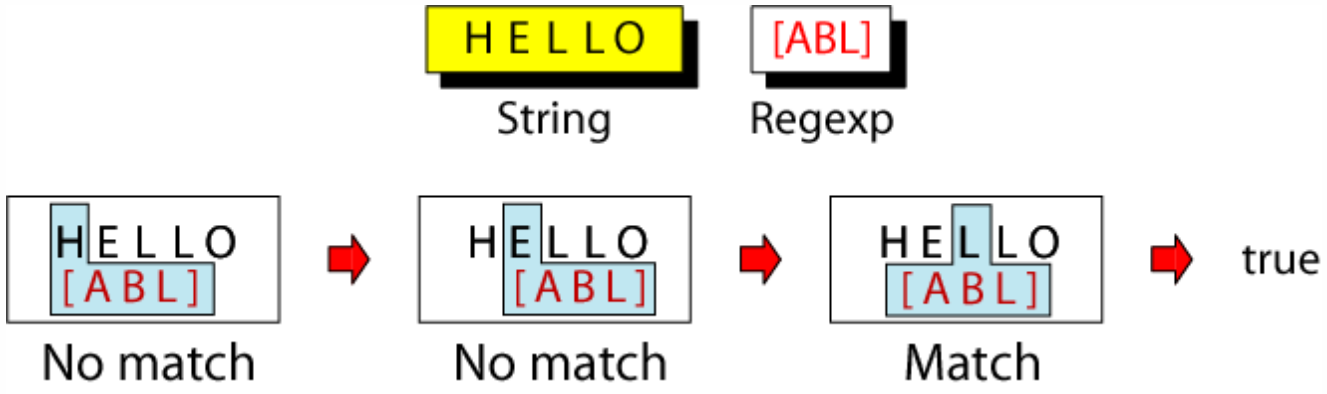
matches any single character except for a new line character (\n)



Class Atom

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

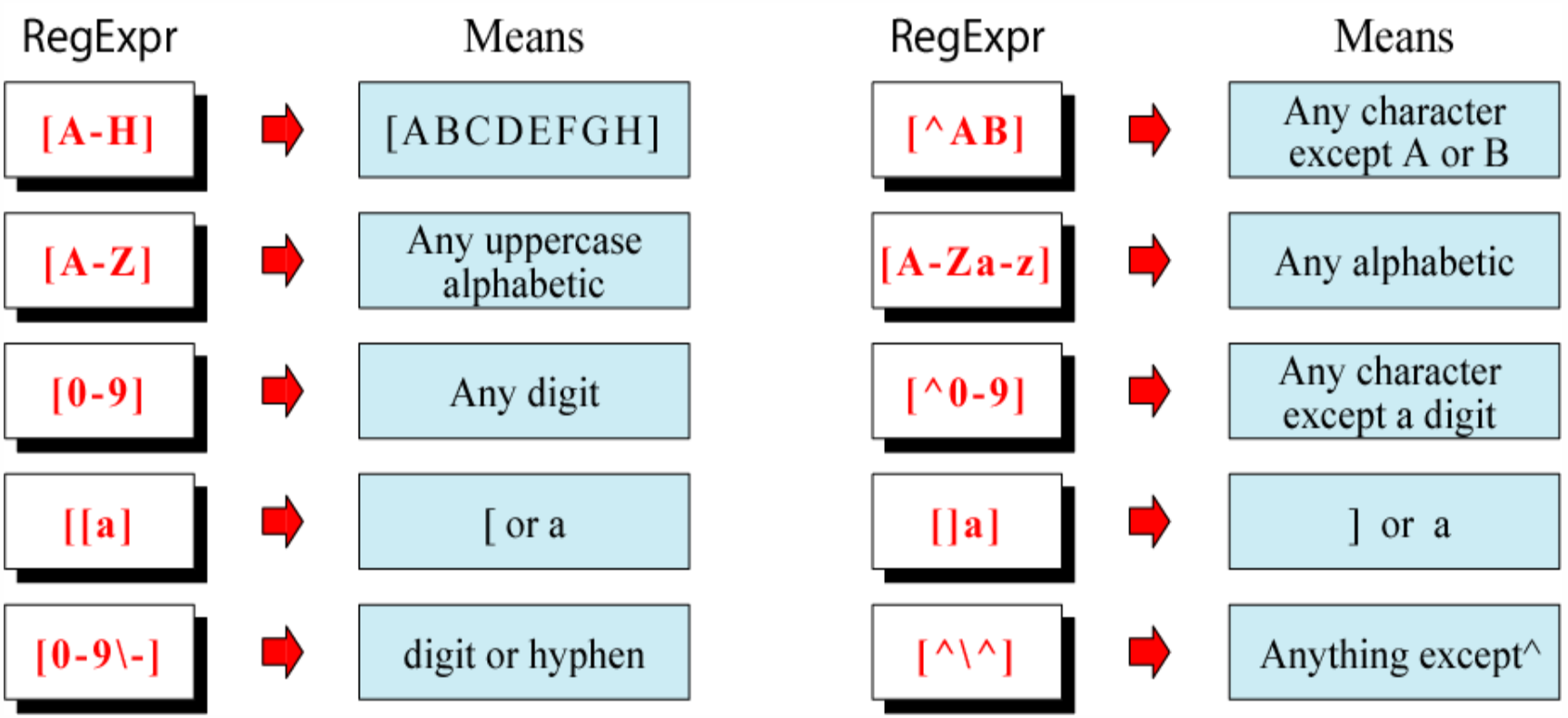
Example: [ABC] matches either A, B, or C.



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



Shortcuts

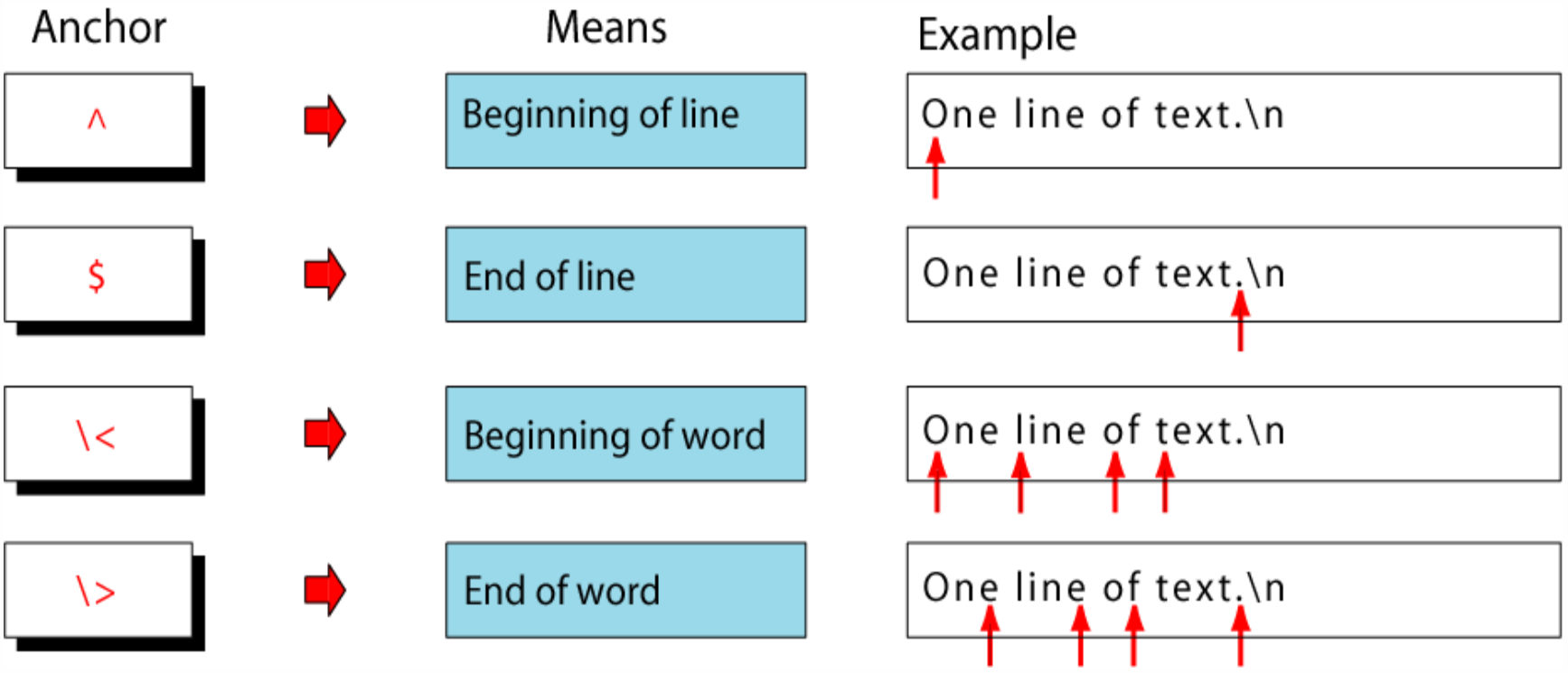
- **\d =>** any digit `[0-9]`
- **\w** => any “word” character `[A-Za-z0-9_]`
- **\s** => any white space `[\t\n\r\f ]`

- **\D** => any character except a digit `[^\d]`
- **\W** => any character except a “word” character `[^\w]`
- **\S** => any character except a white space `[^\s]`

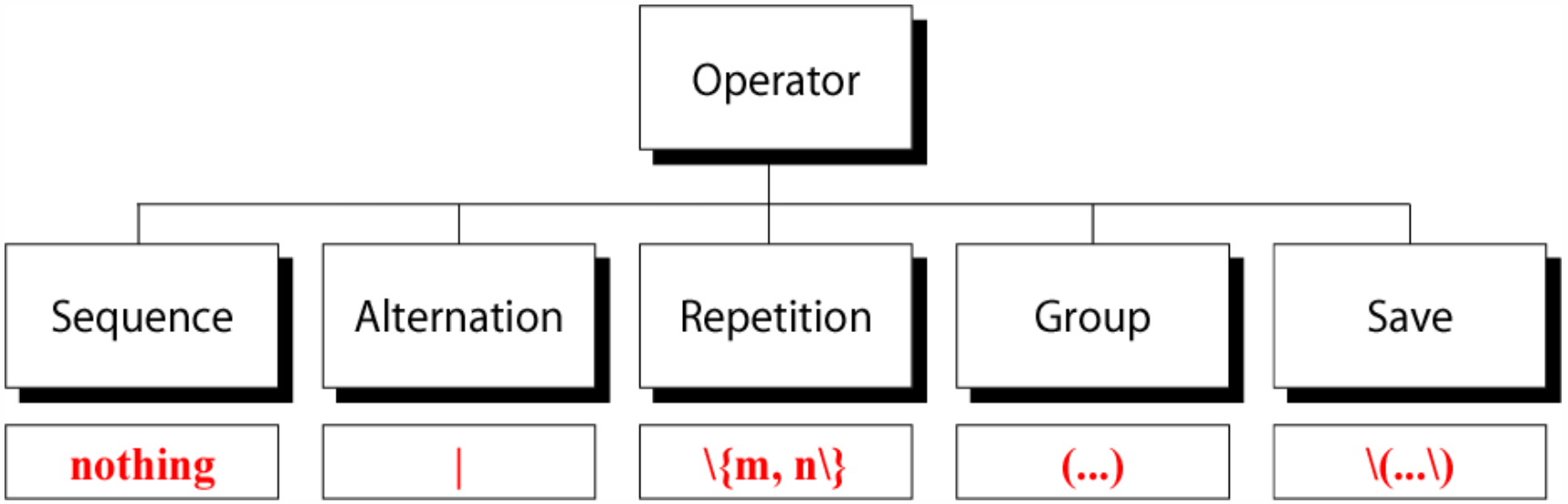
- Can use any of these in conjunction with quantifiers,
- **/\s*/** => any amount of white space

Anchors

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



Operators


Operator
Sequence
Alternation
Repetition
Group
Save
nothing
|
\{m, n\}
(...)
\(...\)

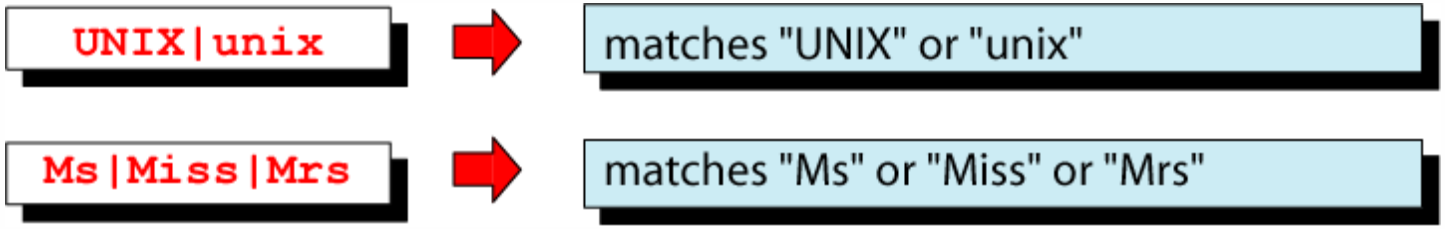
Sequence Operator

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 "-"

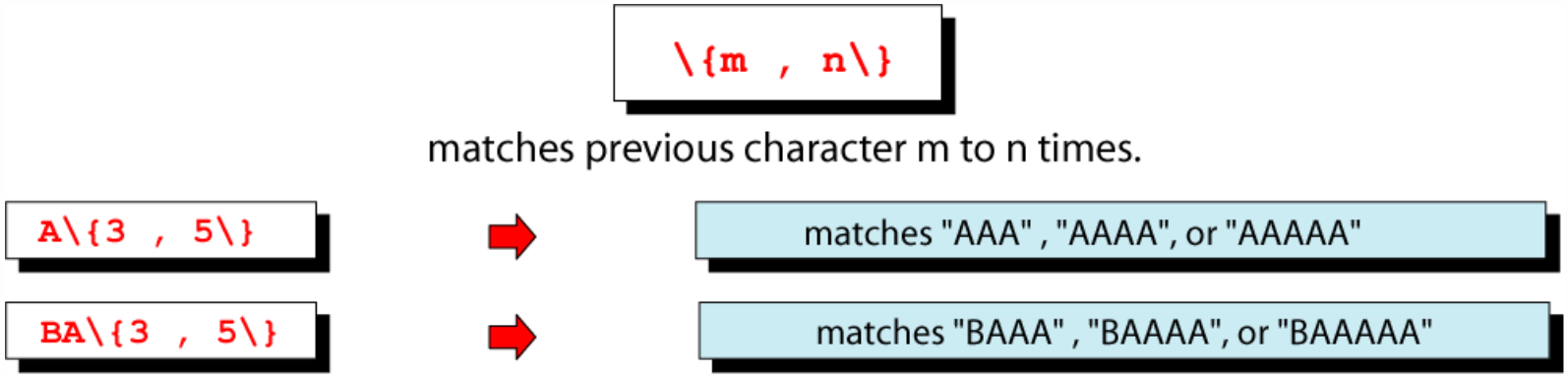
Alternation Operator: | or \|

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



Repetition Operator: \{…\}

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



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`

Univerzitet Singidunum

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.