

U.+?g R[def]gu..r Expres+ions (in | inn) D[a-z]+?!
[Rr]eseach/!

(Using Regular Expressions in Digital Research!)

Kevan A. Feshami
Graduate Assistant, CRDDS

With input from Prof. Vilja Hulden
History Dept, CU Boulder

CRDDS@Colorado.edu

Regular Expressions (Regex), Some Basics

- ▶ Character sequence(s) to create search patterns
- ▶ Uses two types of characters
 - ▶ Literal characters, meaning the actual character represented
 - ▶ A means A, b means b (it's case sensitive!)
- ▶ Metacharacters (not the protagonists in a postmodern novel)
 - ▶ Characters that stand in for something else
 - ▶ * does not mean *, but * does (that's so meta!)

What's it good for?

- ▶ Searching! (and cleaning)
- ▶ And coding, but we'll focus on searching today

Metacharacters

* means zero or more of the preceding literal character

► Ex: $ba^*c = bc; bac; baac;$

b
a
c;
etc.

+ means one or more of the preceding literal character

► Ex: $ba+c = bac; baac; baaaaaac$; but not bc

Question:

- ▶ How could we write a regular expression to match books, but not bks?

Answer: $bo+ks$

More Metacharacters

- will match any character
 - ▶ Ex: `b.c` = `bac`; `bbc`; `bcc`
 - ▶ Does not match multiple characters

Question:

- ▶ How can we complete the regular expression `b__c` to match both `bbbbbbbc` and `baaaaaaac`?
- ▶ Will it match `bbbaadsc`?

Answer: $b * c$ or $b + c$
(and yes it will)

Even *More* Metacharacters

- `*` will, however, match whole lines up to a line break
 - ▶ Ex: `b.*y` for the phrase “by norlin library” will not match “by”; but it will match “by norlin library”

To stop at the first instance of a literal character, use `?`

- ▶ Ex: `b.*?y` for the phrase “by norlin library” will only match “by”; does not match “by norlin library”

Question:

- ▶ How could we write a regular expression to match the phrase “Norlin has lots of books” for the phrase “Norlin has lots of books and a coffee shop, too”? How about just “Norlin has”

Answers: N.+s and N.+?s

Alternatives

| means either/or for literal characters on either side of this metacharacter

- ▶ Ex: `a | b` = `a`; `b`; but not `ab` (will not match both together, just either/or)

Question:

- ▶ How can we write a regular expression that will match `a` and `bbbbbb`?
 - ▶ Will it match `aaaaaaa` and `bbbbbbbbb`?
 - ▶ If not, what would?

Answer: $a \mid b^+$ (or $*$)

No, but $a^+ \mid b^+$ will

Again with the Alternatives

() let you group literal characters within strings in conjunction with **|**

- ▶ Ex: `p(o | i)t`= pot; pit; but not pet, pat, or put

Question:

- ▶ What two ways could we write a regular expression that matches houses or homes?
- ▶ How could we write a regular expression that matches house, home, houses, or homes?

Answer: houses | homes or (house | home)s
(house | home)s*, this will also match
homesssssssss, so use ? if you need to

Character Classes

[] let's you group literal characters together in a class

- ▶ Ex: `[abc]=a; b; c;` but not `abc`
- ▶ `[abc][abc]= ab; bc; ab;` but not `abc`
- ▶ `[abc][abc][abc]= abc; bbc; cbc; abb; etc.`

Question:

- ▶ How could we write a regular expression that doesn't require repeatedly typing a class? (i.e. to get `abc` with typing `[abc][abc][abc]`)

Answers: $[abc]^+$

Character Class, cont'd

You can also add a - to cover a range in the alphabet (or numerical range)

- ▶ [a-z]= a; b; c; d; q; r; s; t; z, etc.
- ▶ [0-9]= 0; 1; 2; 3; 4; 5; 6; 7; 8; 9
- ▶ Or combine them [a-z0-9]= a; 0; b; 1; c; 2, etc.

Character classes can also find matches for capital and lower case letters, i.e. [Ff] or [Ee]

Question:

- ▶ How could we write a regular expression that matches house and House?
- ▶ How about house, House, home, Home, houses, Houses, home, Homes?

Answer: [Hh]ouse
([Hh]ouse | [Hh]ome)s*

One Last T[hij]+ng\!

If you'd like . + * ? and other punctuation to match the actual, literal character (i.e. not the metacharacter), use \

► Ex: \.com= .com; but not xcom, etc.

Additional Resources

Vilja Hulden's Little Regular Expressionist

- ▶ <https://osf.io/he2qa/>

Regular-Expressions.info

- ▶ <https://www.regular-expressions.info/>

RegExr

- ▶ <https://regexr.com/>

Antconc

- ▶ <http://www.laurenceanthony.net/software/antconc/>

Notepad++

- ▶ <https://notepad-plus-plus.org/>

Additional Support

CRDDS Consultation Hours

- ▶ Every Tuesday, 12-2pm, Norlin E206

Scheduled Consultation

- ▶ Email CRDDS@Colorado.edu

Check out other CRDDS events!

- ▶ <https://www.colorado.edu/crdds/events>
- ▶ Sign up for our mailing list!

Kevan A. Feshami

CRDDS, Graduate Assistant