Računalniški praktikum I

Regularni izrazi

Vida Groznik

Regularni izrazi

- Regularni izraz, regex ali regexp je zaporedje znakov, ki definirajo iskalni vzorec.
  Takšen vzorec se ponavadi uporablja v algoritmih za iskanje nizov, pri operacijah „find“ ali „find and replace“ na nizih
  (vir: Wikipedia)

- Različna orodja, ki uporabljajo regex: grep, IDE code competition, Kate, ed, find, locate, vi, emacs, .NET, Java SDK, Exalead, itd.

Osnovni koncepti

- Vsak znak predstavlja samega sebe, **razen** [\^$.|?*+(){}
- \ ki mu sledi *[\^$.|?*+(){}* predstavlja sledeči znak
- *[niz]* predstavlja **en** znak iz niza v oklepajih: *n*, *i* ali *z*
- *[^niz]* predstavlja negacijo niza – znak, ki **ni** *n*, *i* ali *z*
- - (razen pred ali za []) predstavlja razpon: *[a-zA-Z0-9]* vsi znaki in vse številke
- *[-]* predstavlja *minus*
- *(niz)* predstavlja *podniz/podizraz,* ki se ga lahko kasneje prikliče
- | predstavlja *izbiro*:
  a|b pomeni *a ali b*; *a(b|c)d* pomeni *abd ali acd*

Osnovni koncepti

- . ustreza *kateremu koli znaku* (razen predsledku ali novi vrstici):
  a.cd je lahko *abcd, aXcd,...*
- *[.]* predstavlja *piko*
- * za znakom pomeni **nič ali več** ponovitev znaka:
  *ab*c* je lahko: *ac, abc, abbc,...*
  *a(bb)*c* je lahko: *ac, abbc, abbbbc,...*
  *[xyz]** je lahko *“, x, y, zx, zyx,...*
- *+* za znakom pomeni **ena ali več** ponovitev znaka:
  ab+c je lahko: *abc, abbc,...*
  [xyz]+ je lahko: *x, y, zx, zyx,...*
- *?* pomeni, da je prejšnji znak (niz) prisoten ali pa ne: *ab?c* je lahko *ac* ali *abc*

Osnovni koncepti

- *{}* omejitev števila ponovitev prejšnjega znaka (podniza)
  {n} prejšnji znak se ponovi *natanko n-krat*: *a{3}* pomeni *aaa*
  {n,m} prejšnji znak se ponovi *vsaj n-krat in največ m-krat*
  {n,} prejšnji znak se ponovi *najmanj n-krat*
- ^ predstavlja **začetek** niza
- *$* predstavja **konec** niza

Okrajšave

ASCII	Pomen
[A-Z]	Velike črke
[a-z]	Male črke
[A-Za-z]	Velike in male črke
[A-Za-z0-9]	Alfanumerični znaki
[0-9]	Cifre
[A-Fa-f0-9]	Heksadecimalni znaki
[][!"#$%&'()*+,./:;<=>?@\^_`{|}~-]	Ločila
[[\t]]]	Presledek in tabulator
[\t\r\n\v\f]	Znaki za presledke
[\x00-\x1F\x7F]	Kontrolni znaki

Primer

- *(a|b)*ccc*
  predstavlja zaporedja znakov, ki se začnejo s ***katerim koli številom*** ponovitev črke ***a*** in ***katerim koli številom*** ponovitev črke ***b*** in se zaključijo s ***tremi ponovitvami*** črke ***c***.
  - Znak **|** ločuje dve možnosti: *a*|*b* pomeni „*a* ali *b*“
  - Znak ***** pomeni ***nič ali več*** ponovitev izraza pred znakom: *(a|b)** pomeni „katero koli število ponovitev znakov *a* ali *b*“
  - (*niz*): oklepaji omejujejo podzaporedje/niz
  - Na koncu imamo 3 ponovitve črke *c*

ccc	cccccc (narobe!)	bbbbaccc
ababaaccc	cccaababa (narobe!)	acccc

Primer

- (Luis Fonsi)|(luis fonsi)

  Luis Fonsi
  luis fonsi

- (L|l)uis (F|f)onsi

  Luis Fonsi
  Luis fonsi
  luis Fonsi
  luis fonsi

- (a*)b(a*)b(a*)b(a*)

  Vsi nizi *a*-jev in *b*-jev kjer se *b* ponovi natanko trikrat.

Primer

- 0|((1|2|3|4|5|6|7|8|9)(0|1|2|3|4|5|6|7|8|9)*)

  Niz 0 in vsi nizi številk, ki se ne začenjajo z 0.
- 0|([1-9][0-9]*)

  Niz 0 in vsi nizi številk, ki se ne začenjajo z 0.
- [A-Z][a-z]*

  Vsi nizi znakov, ki se začenjajo z veliko črko.
- [A-Da-z]*

  Vsi nizi, ki vsebujejo znake A, B, C in D in male črke. Primer: aaaBfdCDsdfsdAzz.

Vaja

- Na desktopu pojdite v direktorij, ki smo ga ustvarili prejšnjič
- **Ustvarite** nov **direktorij** bbb.txt
- **Premakni se** v sirektorij bbb.txt
- Uporabi urejevalnik besedil **emacs** in ustvari datoteko names.txt
  - Kako? Poskusi vtipkati emacs names.txt
- Preveri kaj je shranjeno v datoteki names.txt

Vaja

- Odpri datoteko names.txt z urejevalnikom besedil **nano** in dodaj ime Vida v datoteko, datoteko shrani in jo zapri.

- Preveri kaj je zapisano v datoteki names.txt

- Uporabi regularne izraze z ukazom **egrep** in poišči vse začetne vrstice v datoteki names.txt.
  - Kako deluje ukaz **egrep**? Kako lahko to ugotovimo?

Vaja

- Poišči vse vrstice v datoteki names.txt, ki se začnejo s črko **M**.
- Poišči vrstice, ki se **začnejo s črko M in končajo s črko a** in imajo med njima poljubno število črk.
- Poišči vrstice, ki se **končajo s črko a**.
- Poišči vrstice, ki se **končajo s črko a** in imajo pred njo vsaj še štiri črke.

Vaja

- Vrstice, ki se končajo s črko **a** in imajo natanko štiri črke.
- Vrstice, ki se začnejo ali s črko **M** ali s črko **L**.
- Napiši besedo **lola** na konec dokumenta names.txt brez uporabe urejevalnika besedil.
- Napiši besedo **LOLA** na konec dokumenta names.txt brez uporabe urejevalnika besedil.
- Prikaži vsebino datoteke na ekran.

Vaja

- Izpiši vse vrstice, ki vsebujejo ime lola, pri čemer je lahko vsaka črka mala ali velika.
- Vrstice, ki se začnejo z Marjan in imajo nič ali več črk za tem.
- Vrstice, ki se začnejo z Marjan in imajo vsaj še eno črko za tem.
- Vrstice, ki se začnejo z Marjan in imajo eno ali nič črk za tem.

Vaja

- Prikaži podrobnen opis vsebine direktorija bbb.txt.
- Prikaži podroben opis vsebine direktorija bbb.txt, kjer imajo rezultati končnico .txt
- Kaj naredi ukaz **locate**?
- Uporabi ukaz locate in regularne izraze in poišči vse datoteke, ki se končajo na .txt.