Säännölliset lausekkeet

Jenna Kanerva & Kai Hakala
Turun yliopisto, Informaatioteknologian laitos
Syksy 2015

Regex

- Säännöllinen lauseke (regular expression, **regex**):
- merkkijonojen etsiminen tekstistä
- määritellään lausekkeen muoto ilman että tarvitsee antaa tarkka merkkijono
- esim. hae kaikki sanat, jotka alkavat alkavat a:lla ja päättyvät numeroon

- unixin *grep* työkalu (*grep -P "sana" tiedosto.txt*)
- hakukoneet (*Turun* $\star$ *kirjasto*)

Regex: Esimerkki

Lämpötilat: 20.9. klo 8:00 +9.8 astetta, tuuli 3 m/s, 20.9. klo 15:00 +11.4 astetta, tuuli 2.5 m/s, 15.10. klo 7:30 +5.1 astetta, tuuli 5 m/s, 15.10. klo 15:00 +5.6 astetta, ilmankosteus 31%, 15.10. klo 20:30 +4.7 astetta, tuuli 1 m/s, 20.10. klo 7:00 -0.2 astetta, tuuli 0 m/s, 20.10. klo 14:30 +2.3 astetta, tuuli 0 m/s

Regex

- Perustuu tila-automaatteihin (käsitellään tarkemmin morfologian yhteydessä)
- Syntaksi vaihtelee hieman riippuen käytetäänkö regexejä esim. Pythonista vai Perlistä

Regex

http://xkcd.com/208/

Regex

- Yksinkertaisin regex:

  grep -P ”sana” tiedosto.txt

  $\rightarrow$ palauttaa kaikki rivit, joilla esiintyy merkkijono *sana*
- Isot ja pienet kirjaimet:

  grep -P ”[Ss]ana” tiedosto.txt

  $\rightarrow$ *sana* tai *Sana*

Regex


- Sallittujen merkkien määrittely:
- **[a-zääö]** = kaikki pienet kirjaimet
- **[a-zääöA-ZÅÄÖ0-9_]** = sekä isot että pienet kirjaimet + numerot + alaviiva (voidaan myös lyhentää \w)
- negaatio: **[^a-zääöA-ZÅÄÖ0-9_]** (tai \W)

  Esim. viime slidessa nähty haku ei ole ihan riittävä, palauttaa myös mm. *xxxSanaxxx*

  → määritellään "sanarajat"

  grep -P "(\W|^)[Ss]ana(\W|$)" tiedosto.txt

Regex

- Merkkien määrä:
  - **a** = yksi a
  - **a?** = ei yhtään tai yksi a
  - **a+** = yksi tai useampi a
  - **a*** = nolla tai useampi a
  - **a{n}** = n kappaletta a:ta

- Esim. **grep -P "sanat?" tiedosto.txt**

  $\rightarrow$ *sana* tai *sanat*

Regex

- Kaikki *sana* alkavat sanat, jotka jatkuvat jollain kirjaimella (pelkkä *sana* ei kelpaa):

  grep -P "sana[a-zääö]+" tiedosto.txt

  → *sana* + vähintään yksi kirjain

  → esim. *sanassa*, *sanakirja*

- Kaikki *sana* alkavat sanat, joiden pääte on kolme kirjainta pitkä (esim. *sanassa*):

  grep -P "sana[a-zääö]{3}(\W |$)" tiedosto.txt

  → *sana* + 3 muuta kirjainta

  → esim. *sanassa*, *sanaksi*

Regex

- Sähköpostiosoitteiden poimiminen tekstistä:

  grep -P "([a-zåäöA-ZÅÄÖ]+\.)*

  → etunimi. (nolla tai useampi esiintymä näitä)

  [a-zåäöA-ZÅÄÖ0-9_]+

  → sukunimi, joka voi sisältää numeroita ja alaviivoja (vähintään yksi merkki)

  @[a-zåäöA-ZÅÄÖ]+\.[a-zåäöA-ZÅÄÖ]{2,4} (\W|$)" tiedosto.txt

  → @utu.fi (yksi esiintymä näitä, viimeisen merkkijonon täytyy olla 2-4 merkkiä pitkä)

  Esim. *etunimi.sukunimi@utu.fi*, *nimi@utu.fi*, *nimi_1@gmail.com*

Regex


- Etsi ja korvaa:
- Helpoin kirjoittaa yksinkertainen Perl-komento
- Perus syntaksi:

  cat tiedosto.txt | perl -pe "s/alkuperäinen/korvaava/g"
- jossa **g** tarkoittaa globaalia korvausta (muuta kaikki esiintymät, ei vain rivin ensimmäistä)

Regex

- Korvaa kaikki *organise* sanat *organize* sanoilla:
  cat tiedosto.txt | perl -pe "s/organise/organize/g"

Regex

- Regexien käyttö Pythonissa:

```
import re

regex=re.compile(u"\b(?:[a-z]+\.)*[a-z0-9_]+@[a-z]+\.[a-z]{2,4}\b")

print regex.findall(u"sähköpostiosoitteita: etunimi.sukunimi@utu.fi,
nimi@utu.fi, nimi_1@gmail.com ja nimi.nimi.nimi@gmail.com")

[u'etunimi.sukunimi@utu.fi', u'nimi@utu.fi',
u'nimi_1@gmail.com', u'nimi.nimi.nimi@gmail.com']
```

Regex

Pythonin regex manuaali

Python Regex Tool

Regex

- Regexien käyttötarkoituksia:
  - Tiedon poimiminen tekstistä
  - Käyttäjäsyötteen oikeellisuuden tarkastaminen
  - search and replace
  - Tokenisoija $\rightarrow$ Kai puhuu tästä lisää!
  - Tietynlaisten sanojen/tagien esiintyminen yhdessä
  - sähköpostispämmäys