KI/OPS

REGULÁRNÍ VÝRAZY

Regulární výrazy

- **Regulární výrazy jsou jazyk pro popis jazyků a manipulaci s textem.**
  - Jazyk - množina řetězců
    - Konečná: Petr,Pavel
    - Nekonečná: všechny řetezce začínající na a
  - Terminály – znaky mající význa sebe sama
  - Neterminály (metaznaky)
    - popisují znakovou množinu: [:digit:]
    - Kotvy: ^,$
    - Operace: *,{n}
    - Odkazy !!!: grep -e '^\(0*\)a\1$' <<< "00a00"
    - Aby neterminál měl význam terminálu je typicky přidat escape sekvence \$

Matoucí poznámka: Poslední příklad není dle definice (T.I.) regulární (zpětné odkazy), ale moderní nástroje to dovolí popsat.

Základní charakteristika

- **Existuje více druhů RE**
  - Základní regulární výrazy-BRE
  - Rozšířené regulární výrazy-ERE
  - Perlovské regulární výrazy-PRE
- **Odlišnost**
  - Co je terminál a neterminál
  - Podpora znakových sad
  - Zpětných odkazů
- **Nástroje**
  - grep, egrep (grep -E) -BRE,ERE, PRE-experimentálně
  - C#- třída Regex
  - Java – třída Pattern
  - C, C++ - knihovna PCRE

Regulární výrazy I

Znak	Význam	
terminál	řetězec tvořený daným znakem-terminálem	a
.	řetězec tvořený libovolným znakem (jediným)	., \.
[znakovámnožina]	řetězec tvořený právě jedním znakem ze znakové množiny	[a-d,p,k,q]
[^znakovámnožina]	řetězec tvořený právě jedním znakem neobsaženým ve znakové množině	[^a-d,p,k,q]
*, \\, \^, \$, \ +, \?, \\|,\[, \],\(, \), \ {, \}	Terminály v ERE (některé i v BRE)	

Opakovač	Opakování	
*	0..nekonečno	.*, a*.
+	1..nekonečno	(ab)+
?	0,1	\-?[[:digit:]]+
{n,m},{n,},{n}	n-m,n-nekonečno, právě n	[0,1]{4}

Regulární výrazy II

- **Znakové množiny**

Znaková množina	význam
[:digit:]	číslice
[:alpha:]	Písmenné znaky
[:alnum:]	[:digit:] + [:alpha:]
[:lower:]	malá písmena
[:upper:]	velká písmena

- **Kotvy pozice v řetězci ^,$,\b**
  - Začátek, resp. konec řádku (pro grep důležité)
  - \b hranice, počátky slov
    - grep -E '\b[0,1]{4}\b' <<<"a0000a"
    - grep -E '^\-?[[:digit:]]+$' <<<"123"
- **Operátor volby |**
    - grep -E '^(petr|pavel)$'

Regulární výrazy |||

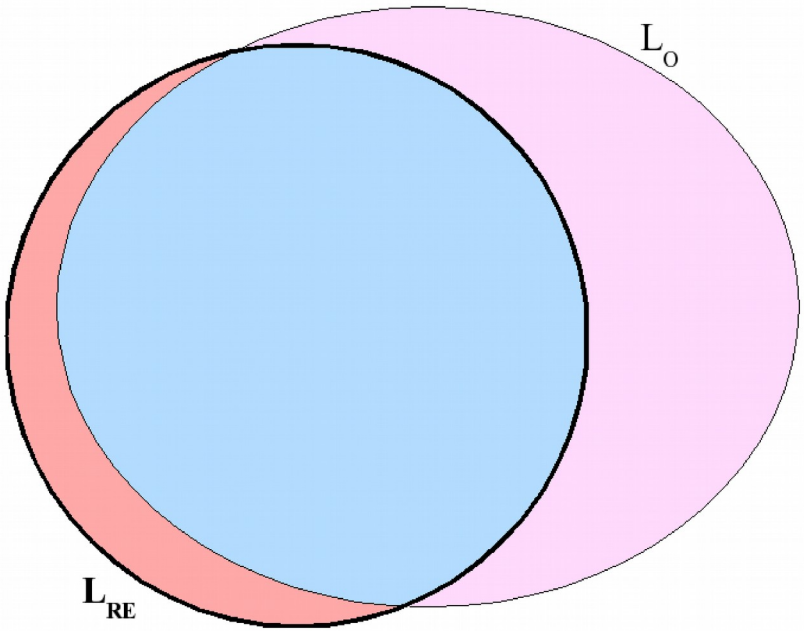
- **Příklady**

Rodné číslo (oba KO):

```
grep -E '^[[:digit:]]{6}\/[[:digit:]]{3,4}$'
grep -E '^[[:digit:]]{2}([0,1]|[5,6])[[:digit:]][0,1,2,3][[:digit:]]\/[[:digit:]]{3,4}$'
```

Čísla v C (C#):

```
[+-]?([0-9]+(\.[0-9]*)?|\.[0-9]+)([eE][+-]?[0-9]+)?
```



Lo – očekáváný jazyk

LRE – to co jsem popsal

- Je náročné ověřit, že Lo=LRE
- Komplikovaný popis jak má Lo vypadat

  grep -E '\b([[:alpha:]]+[[:digit:]]{3})*\b'

Nástroje

- **Grep – filtruje řádky obsahující daný RE**
  - Volba typů BRE (-G),ERE (-E), PRE (-P)
  - Lze vnutit hledání řetězců místo RE (-F)
  - Prohledává soubory
    - grep -r -Eni 'WriteLine' ~/Tmp/
- **Sed – editor pro proudové zpracování textů**
  - ls -l | sed -e 's/kubera/ANONYM/g'

    cvičení !!!