THEORETICAL COMPUTER SCIENCE
Jagiellonian University

Wyrażenia regularne i grep

Wyrażenie regularne to sekwencja znaków opisująca wzorzec tekstowy, czyli zbiór słów odpowiadających podanemu opisowi.

Program `grep` służy do wyszukiwania wzorców tekstu w tekście. Typowe zastosowanie to wyszukiwanie małego słowa/wzorca w wielkim pliku tekstowym. *Grep* jest w systemie *Unix* prawie od początku, pierwsza wersja powstała w 1973 roku. Kilka opcji wywołania programu `grep` omawiamy na końcu zestawu. Tutaj przedstawiamy jedynie typowe wywołanie:

`grep 'krowa' tekst.in` (drukuje wszystkie linie w pliku *tekst.in* zawierające tekst *krowa*)

W powyższym przykładzie `krowa` jest wyrażeniem regularnym. Na wyrażenie regularne składają się znaki trzech różnych typów: *znaki pozycjonujące*, *zbiory znaków* oraz *znaki modyfikujące*.

Znaki pozycjonujące to:

- `^` początek linii
- `\$` koniec linii
- `\<` początek słowa
- `\>` koniec słowa

Powyższe znaki pełnią funkcję pozycjonujących tylko wtedy, gdy są w odpowiednim miejscu wyrażenia regularnego.

Przykłady:

- `^A` „A” na początku linii
- `A$` „A” na końcu linii
- `A^` „A^” gdziekolwiek w linii
- `$A` „$A” gdziekolwiek w linii
- `^^` „^” na początku linii
- `$$` „$” na końcu linii

Najprostszym wyrażeniem regularnym są ciągi liter. Dla przykładu wyrażenie „a” odpowiada po prostu literze *a*, natomiast wyrażenie „krowa” odpowiada słowu *krowa*.

Zbiory znaków (z przykładami użycia):

`.`	linia, w której występuje dowolny znak
`^.$`	linia zawierająca dokładnie jeden znak
`^[0123456789]$`	linia zawierająca dokładnie jedną cyfrę
`^[0-9]$`	to samo, co wyżej, ale krócej zapisane
`[A-Za-z0-9_]`	linia z co najmniej jedną literą, cyfrą lub podkreślnikiem
`^T[a-z][aeiou]`	linia zaczynająca się od T, potem dowolnej litery i samogłoski łacińskiej
`[]`	linia zawierająca nawiasy kwadratowe
`[0]`	linia zawierająca 0
`[^0-9]`	linia zawierająca znak, który nie jest cyfrą
`[-0-9]`	linia zawierająca znak, który jest cyfrą lub minusem
`[0-9-]`	to samo, co wyżej
`[^-0-9]`	linia zawierająca znak, nie który jest cyfrą ani minusem
`[]0-9]`	linia zawierająca znak, który jest cyfrą albo]
`[0-9]]`	linia zawierająca cyfrę, a po niej]
`[0-9-z]`	linia zawierająca cyfrę lub znak z zakresu od 9 do z
`[0-9\-a\]]`	linia zawierająca cyfrę, minus, „a” lub]

Istnieją również alternatywne oznaczenia dla zbiorów - klasy znaków `POSIX`:

`[:alnum:]`	`A-Za-z0-9`
`[:alpha:]`	`A-Za-z`
`[:blank:]`	spacja lub tabulator
`[:cntrl:]`	znaki kontrolne
`[:digit:]`	`0-9`
`[:graph:]`	znaki graficzne, czyli znaki o kodach `ACSII` 33-126
`[:lower:]`	`a-z`
`[:print:]`	znaki drukowalne (znaki graficzne + spacja)
`[:space:]`	białe znaki
`[:upper:]`	`A-Z`
`[:xdigit:]`	`0-9A-Fa-f`

Uwaga! Te klasy wymagają ujęcia w dodatkową parę nawiasów kwadratowych, np.:

`[[:digit:]]`	zbiór oznaczający dowolną cyfrę
`[[:alnum:]_]`	zbiór oznaczający dowolną literę, cyfrę bądź podkreślnik

Znaki modyfikujące:

`\`	zmienia znaczenie następnego znaku ze specjalnego na zwykły i na owdrót
`*`	następując po znaku (lub wyrażeniu w nawiasach) sprawia, że ten występuje 0 lub więcej razy
`\+`	następując po znaku (lub wyrażeniu w nawiasach) sprawia, że ten występuje 1 lub więcej razy
`\?`	następując po znaku (lub wyrażeniu w nawiasach) sprawia, że ten występuje 0 lub 1 raz

Co więcej, można nakazać danej sekwencji znaków pojawiać się dokładnie określoną liczbę razy przy pomocy wstawienia przedziału wewnątrz `\{` i `\}`.

Przykłady:

`*`	dowolna linia z gwiazdką
`\*`	to samo, co wyżej
`\\`	dowolna linia z backslashem
`^*`	dowolna linia zaczynająca się od gwiazdki
`^A*`	dowolna linia
`^A\*`	dowolna linia zaczynająca się od A*
`^AA*`	dowolna linia zaczynająca się od A
`^A*B`	dowolna linia zaczynająca się od B lub jednym lub więcej A, poprzedzającym B
`^AA*B`	dowolna linia zaczynająca się jednym lub więcej A, poprzedzającym B
`^A\+B`	to samo, co wyżej
`^A\?B`	dowolna linia zaczynająca się od A lub od AB
`^A\{4,8\}B`	dowolna linia zaczynająca się od 4, 5, 6, 7 lub 8 A, poprzedzających B
`^A\{4,\}B`	dowolna linia zaczynająca się od co najmniej 4 A, poprzedzających B
`^A\{4\}B`	dowolna linia zaczynająca się od AAAAB
`^\{4,8\}`	dowolna linia zaczynająca się od 4,8
`A{4,8}`	dowolna linia z A4,8

Istnieje również wzorzec pozwalający wyszukiwać słowa z powtórkami. Przykładowo, wyrażenie `[a-z][a-z]` wskaże nam dwie sąsiednie małe litery, ale niekoniecznie takie same (podobnie jak `[a-z]*` wskaże nam dowolnej długości ciąg małych liter, lecz niekoniecznie różnych). Wyjściem jest zastosowanie nawiasów: `\(\)` i `\)`, w które zamykamy interesujący nas podwzorzec, by później odwołać się do niego przy pomocy cyfry poprzedzonej backslachem. Dysponujemy 9 takimi slotami pamięci, oznaczonymi od `\1` od `\9`. Przykładowo, pięcioliterowych palindromów (takich jak np. „radar") możemy szukać przez następujące wyrażenie regularne:

```
\([a-z]\)\([a-z]\)[a-z]\2\1
```

Wyrażenia regularne wykorzystywane w `grep`ie pozwalają na używanie alternatywy słów, a nie jedynie alternatywy znaków. W przypadku wyszukiwania linii ze słowami „alfa" lub „beta" możemy postąpić tak, jak poniżej:

```
grep '\(alfa\|beta\)' tekst.in
```

Podamy teraz kilka przydatnych opcji polecenia `grep`:

`grep -c '^[A-Z]' tekst.in`	wypisuje liczbę linii zawierających podany wzorzec
`grep -i 'ErRoR' tekst.in`	wypisuje linie zawierające słowo „error", „ERROR", „eRrOr" itd. (parametr `-i` sprawia, że `grep` nie rozróżnia pomiędzy małymi a dużymi literami)
`grep -n '^[A-Z]' tekst.in`	wypisuje linie zawierające podany wzorzec wraz z numerami tych linii w pliku
`grep -v '^[A-Z]' tekst.in`	wypisuje linie, w których nie występuje wzorzec (w tym przypadku linie, które nie zaczynają się od dużej litery)
`grep -f wzorce.in tekst.in`	wypisuje linie zawierające wzorce z pliku „wzorzec.in" (jeden wzorzec musi zawierać się w jednej linii)
`grep -F lista tekst.in`	dzieli wzorzec „lista" na ciąg wozrców (muszą być oddzielone znakiem nowej linii) i wypisuje linie „tekst.in" zawierające te wzorce

Najpopularniejszym zastosowaniem polecenia `grep`, oprócz wyszukiwania wzorca w pojedynczym pliku tekstowym, jest przeszukiwanie zawartości wszystkich plików w podanym katalogu (rekurencyjnie): `grep -r "wzorzec" Projekt` wypisze informacje o wszystkich wystąpieniach „wzorca" w katalogu „Projekt" i jego podkatalogach.