Η γλώσσα PERL

Παντελής Μπάγκος

Παν/μιο Θεσσαλίας

2025

Δομές Ελέγχου/Διαχείριση IO

- `if/unless`
- `while/until`
- `do{}while/until`
- `for`
- `foreach`
- `last, next, redo, labels`

if/unless

```
if (some condition)
{
}
elsif
{
}
else
{
}
```

conditions

- False : “0”, “”
- True: otherwise

while/until

```
while (condition)
{
…
}

until (condition)
{
…
}
```

do {}while/until

```
do
{
  …
} while (condition)

do
{
  …
} until (condition)
```

for

```
for (initialization; condition;update)
{
  …
}

for ($i=1; $i<=10;$i++)
{
  print “$i\n”;
}
```

foreach

```
foreach $i(@list)
{
  …
}

@a=(1,2,3,4,5);
foreach $i(@a)
{
  print “$i\n”;
}
```

last

```
while (condition 1)
{
…
  if(condition 2)
  {
  …
  last;
  }
}
#
```

last

Ισχύει μόνο για:

`for, foreach, while, until`

next

```
while (condition 1)
{
…
  if(condition 2)
  {
  …
  next;
  }
#
}
```

redo

```
while (condition 1)
{
#
…
  if(condition 2)
  {
  …
  redo;
  }
}
```

LABELS

```
L1:for ($i=1; $i<=10;$i++){
    L2:for ($j=1; $j<=10;$j++)
   {
       if($i$j==63)
       {
             print “$i *$j =63\n”;
             last L2;
       }
       if ($j>=$i)
       {
             next L2;
       }
   }
}
```

STDIN

```
$a=<STDIN>;
chomp ($a);

chomp ($a=<STDIN>);
```

<>

```
while(<>)
{
  print $_;
}
```

Εκτελέστε:

```
Perl program.pl file
```

Input-Output

- open FILEHANDLE, “filename”;
- `open IN, “/etc/passwd”;`
- `$x=<IN>;`
- `print $x;`
- `close IN;`
- `open OUT, “>(>)tempfile”;`
- `print OUT “bla bla bla\n”;`

@ARGV

- `perl program.pl file1 file2 …`
- `file1: $ARGV[0]`
- `file2: $ARGV[1]`
- …

open (IN, "filename")	Ανοίγει ένα υπάρχον αρχείο με το ονομα filename και ονομαζει το διαχειριστή, IN
open (OUT, "> filename")	Δημιουργεί ένα νέο αρχείο με το όνομα filename και γράφει σε αυτό χρησιμοποιώντας το όνομα διαχειριστή OUT
open (IN, ">> filename")	Προσθέτει στο τέλος ενός αρχείου με το όνομα filename και χρησιμοποιεί το όνομα διαχειριστή OUT

Πίνακας 12.6: *Τρόποι με τους οποίους συντάσσεται η εντολή open*

Κανονικές εκφράσεις

Με τις κανονικές εκφράσεις (regular expressions) μπορούμε να ελέγξουμε εάν μια συμβολοσειρά είναι ίδια με κάποια άλλη, ή περιέχει ένα συγκεκριμένο μοτίβο. Η λειτουργία αυτή είναι ιδιαίτερα χρήσιμη καθώς όπως είδαμε στο κεφάλαιο 4, τα μοτίβα είναι ιδιαίτερα διαδεδομένα στην ανάλυση αλληλουχιών στη βιοπληροφορική. Οι κανονικές εκφράσεις οροθετούνται από τις καθέτους (/) και περιέχουν μια ακολουθία από χαρακτήρες που πρέπει να ταιριάξουν με τους χαρακτήρες μέσα στο σώμα μιας συμβολοσειράς. Για παράδειγμα η εντολή:

```
$dna=~/GAATTC/;
```

ελέγχει αν στη συμβολοσειρά $dna περιέχεται η αλληλουχία GAATTC η οποία αντιστοιχεί σε θέση δράσης μιας περιοριστικής ενδονουκλεάσης. Όταν σε μια συνθήκη βρούμε κάτι σαν /GAATTC/, έχουμε μια απλή συνθήκη ταιριάσματος συμβολοσειρών. Επίσης είναι δυνατό μέσω των κανονικών εκφράσεων όταν βρεθεί μια ακολουθία μέσα σε μια συμβολοσειρά να αντικατασταθεί με κάτι άλλο, με την απλή εντολή /GAATTC/CTTAAG/ (σε αυτή την περίπτωση αντικαθιστούμε το GAATTC με CTTAAG).

Μεταχαρακτήρας	Περιγραφή
.	Οποιοσδήποτε χαρακτήρας εκτός από την αλλαγή γραμμής
^	Αρχή μιας γραμμής
$	Τέλος μιας γραμμής
\w	Οποιοσδήποτε χαρακτήρας λέξης
\W	Οποιοσδήποτε χαρακτήρας εκτός από χαρακτήρα λέξης
\s	Χαρακτήρας διαστήματος
\S	Οποιοσδήποτε χαρακτήρας εκτός από χαρακτήρα διαστήματος
\d	Οποιοδήποτε ψηφίο
\D	Οποιοσδήποτε χαρακτήρας εκτός από ψηφίο

Πίνακας 12.7: *Μεταχαρακτήρες κανονικών εκφράσεων*

Εξ' ορισμού, οποιοσδήποτε χαρακτήρας ή μεταχαρακτήρας σε μια κανονική έκφραση ταιριάζει ακριβώς μια φορά. Μια αναζήτηση με κανονική έκφραση μπορεί να προσπαθεί να ταιριάξει όσο το δυνατόν περισσότερες φορές μέσα στο «κείμενο» την ακολουθία μας, ή όσο το δυνατόν λιγότερες αντίστοιχα. Με την τοποθέτηση ενός ποσοδείκτη (Quantifier) μετά το χαρακτήρα, η Perl μπορεί να ταιριάξει το χαρακτήρα αυτό για συγκεκριμένο αριθμό επαναλήψεων, ή και διάστημα επαναλήψεων. Ο απλούστερος ποσοδείκτης είναι ο {n}, ο οποίος προσπαθεί να ταιριάζει το πρότυπο ακριβώς n φορές. Για παράδειγμα η εντολή:

```
$sequence=~/AAAT{5}CCG/;
```

ελέγχει αν η συμβολοσειρά $sequence ταιριάζει στην αλληλουχία AAATTTTTCCG (δηλαδή, ελέγχει το πρότυπο PROSITE A-A-A-T(5)-C-C-G). Προσέξτε, ότι αυτή είναι μια έκφραση που μπορεί να έχει αληθείς (true) ή ψευδείς (false) τιμες, και κατά συνέπεια θα πρέπει να ελεγχθεί, συνήθως σε κάποια δομή if. Φυσικά, όπως είδαμε στο κεφάλαιο 4, όλες οι εκφράσεις PROSITE αντιστοιχούν σε μια κανονική έκφραση, οπότε, μπορούμε να χρησιμοποιήσουμε αυτούσιους τους κανόνες που είδαμε εκεί για να πραγματοποιήσουμε αναζητήσεις. Στον Πίνακα 12.8 παρουσιάζονται οι ποσοδείκτες κανονικών εκφράσεων.

Regular Expressions Syntax

char	meaning
^	beginning of string
$	end of string
.	any character except newline
*	match 0 or more times
+	match 1 or more times
?	match 0 or 1 times; or: shortest match
|	alternative
( )	grouping; “storing”
[ ]	set of characters
{ }	repetition modifier
\	quote or special

Regular Expressions Syntax

Matching

\w matches any single character classified as a “word” character
\W matches any non-“word” character
\s matches any space character
\S matches any non-space character
\d matches any digit character, equiv. to [0-9]
\D matches any non-digit character

Examples

Expression	Matches...
abc	abc (that exact character sequence, but anywhere in the string)
^abc	abc at the *beginning* of the string
abc$	abc at the *end* of the string
a|b	either of a and b
^abc|abc$	the string abc at the beginning or at the end of the string
ab{2,4}c	an a followed by two, three or four b's followed by a c
ab{2,}c	an a followed by at least two b's followed by a c
ab*c	an a followed by any number (zero or more) of b's followed by a c
ab+c	an a followed by one or more b's followed by a c
ab?c	an a followed by an optional b followed by a c; that is, either abc or ac
a.c	an a followed by any single character (not newline) followed by a c
a\.c	a.c exactly
[abc]	any one of a, b and c
[Aa]bc	either of Abc and abc
[abc]+	any (nonempty) string of a's, b's and c's (such as a, abba, acbabcacaa)
[^abc]+	any (nonempty) string which does *not* contain any of a, b and c (such as defg)
\d\d	any two decimal digits, such as 42; same as \d{2}
\w+	a "word": a nonempty sequence of alphanumeric characters, such as foo and 12bar8 and foo_1

Ποσοδείκτης	Περιγραφή
?	0 ή 1 εμφάνιση (είναι σημαντικός, γιατί αναγκάζει το πρότυπο να μην είναι «άπληστο»)
+	1 ή περισσότερες εμφανίσεις
.	0 ή περισσότερες εμφανίσεις
{n,m}	Μεταξύ n και m εμφανίσεων
{n, }	Τουλάχιστον n εμφανίσεις
{ ,m}	Όχι περισσότερες από m εμφανίσεις

Πίνακας 12.8: *Ποσοδείκτες κανονικών εκφράσεων*

```
while (<>)
{
      if ($_=~/^AC\s{3}(.*?)\;/)
      {
          print ">$1\n";
      }
      if ($_=~/^\s{5}(.*)/)
      {
          $sequence=$1;
          $sequence=~s/\s//g;
          print "$sequence\n";
      }
}
```

```
@aa = (A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y);
for ( $i=0; $i<500; $i++ ) {
  print '>Random', "$i\n";
      for( $j=0; $j<200; $j++ ) {
            $r = $aa[ int (rand 20)];
            print $r;
            print "\n" if ($j+1)%60 == 0 and $j;
       }
  print "\n";
}
```

```
while (<>){
      if    ($_=~/^>(.*)/)
      {
            $name=$1;
            $seq=<>;
            if($seq=~/(.*LA[GA]C)/)
            {
                  $x=length($1);
                  $a=$a+1;
            }
      }
}
print "$a LIPOPROTEINS FOUND";
```