# **SAS'9** – Perl Regular Expressions Tip Sheet

# **Functions and Call Routines**

#### regex-id = prxparse(perl-regex)

Compile Perl regular expression *perl-regex* and return *regex-id* to be used by other PRX functions.

#### pos = prxmatch(regex-id | perl-regex, source)

Search in *source* and return position of match or zero if no match is found.

#### new-string = prxchange(regex-id | perl-regex, times, old-string)

Search and replace *times* number of times in *old-string* and return modified string in *new-string*.

#### call prxchange(regex-id, times, old-string, newstring, res-length, trunc-value, num-of-changes) Same as prior example and place length of result in res-length, if result is too long to fit into new-string, trunc-value is set to 1, and the number of changes is placed in num-of-changes.

# text = prxposn(regex-id, n, source) After a call to prxmatch or prxchange, prxposn

return the text of capture buffer n.

# call prxposn(regex-id, n, pos, len)

After a call to **prxmatch** or **prxchange**, **call prxposn** sets **pos** and **len** to the position and length of capture buffer **n**.

# call prxnext(regex-id, start, stop, source, pos, len) Search in source between positions start and stop. Set pos and len to the position and length of the match. Also set start to pos+len+1 so another search can easily begin where this one left off.

## call prxdebug(on-off)

Pass 1 to enable debug output to the SAS Log. Pass 0 to disable debug output to the SAS Log.

# call prxfree(regex-id)

Free memory for a *regex-id* returned by **prxparse**.

# **Basic Syntax**

Character	Behavior
//	Starting and ending regex delimiters
	Alternation
()	Grouping

#### Wildcards/Character Class Shorthands

Character	Behavior
•	Match any one character
\w	Match a word character (alphanumeric
	plus "_")
\W	Match a non-word character
\s	Match a whitespace character
\s	Match a non-whitespace character
\d	Match a digit character
\D	Match a non-digit character

#### **Character Classes**

Character	Behavior
[]	Match a character in the brackets
[ ^ ]	Match a character not in the brackets
[a-z]	Match a character in the range a to z

# **Position Matching**

Character	Behavior
^	Match beginning of line
\$	Match end of line
\b	Match word boundary
\B	Match non-word boundary

# **Repetition Factors**

(greedy, match as many times as possible)

Character	Behavior
*	Match 0 or more times
+	Match 1 or more times
?	Match 1 or 0 times
{ n }	Match exactly n times
{n,}	Match at least n times
{n,m}	Match at least n but not more than m
	times

# **Advanced Syntax**

Character	Behavior
non-meta	Match character
character	
{ } [ ] ( ) ^	Metacharacters, to match these
\$. *+?\	characters, override (escape) with \
\	Override (escape) next metacharacter
\ <i>n</i>	Match capture buffer <i>n</i>
(?:)	Non-capturing group

# **Lazy Repetition Factors**

(match minimum number of times possible)

Character	Behavior
*?	Match 0 or more times
+?	Match 1 or more times
??	Match 0 or 1 time
{n}?	Match exactly n times
{n,}?	Match at least n times
{n,m}?	Match at least n but not more than m
	times

#### Look-Ahead and Look-Behind

Character	Behavior
(?=)	Zero-width positive look-ahead
	assertion. E.g. regex1 (?=regex2),
	a match is found if both regex1 and
	regex2 match. regex2 is not
	included in the final match.
(?!)	Zero-width negative look-ahead
	assertion. E.g. regex1 (?!regex2),
	a match is found if regex1 matches
	and regex2 does not match. regex2
	is not included in the final match.
(?<=)	Zero-width positive look-behind
	assertion. E.g. (?<=regex1) regex2,
	a match is found if both regex1 and
	regex2 match. regex1 is not
	included in the final match.
(? )</th <th>Zero-width negative look-behind</th>	Zero-width negative look-behind
	assertion.

# **SAS'9** – Perl Regular Expressions Tip Sheet

# **Basic Example**

## **Data Validation**

```
data phone numbers;
 length first last phone $ 16;
  input first last phone & $16.;
datalines;
Thomas Archer
                  (919)319-1677
                  800-899-2164
Lucy Barr
                  (508) 852-2146
Tom Joad
Laurie Gil
                  (252)152 - 7583
data invalid:
 set phone numbers;
 where not
 prxmatch("/\([2-9]\d\d\) ?" ||
           "[2-9]\d\d-\d\d\d'", phone);
run;
proc sql; /* Same as prior data step */
  create table invalid as
  select * from phone numbers
 where not
 prxmatch("/\([2-9]\d\d\) ?" ||
           "[2-9]\d\d-\d\d\d\d'", phone);
quit;
Output:
  Obs
          first
                    last
                                phone
                            800-899-2164
         Lucy
                    Barr
```

Gil

(252)152-7583

Laurie

# **Search and Replace #1**

```
data _null_;
    input;
    _infile_ =
        prxchange('s/</&lt;/', -1, _infile_);
    put _infile_;
datalines;
x + y < 15
x < 10 < y
y < 11
;

Output:
    x + y & lt; 15</pre>
```

# Search and Replace #2

x < 10 &lt; y

y < 11

```
data reversed names;
  input name & $32.;
datalines;
Jones, Fred
Kavich, Kate
Turley, Ron
Dulix, Yolanda
data names;
 set reversed names;
 name = prxchange('s/(\w+), (\w+)/$2 $1/',
                   -1, name);
run;
proc sql; /* Same as prior data step */
  create table names as
  select
     prxchange('s/(\w+), (\w+)/$2 $1/',
               -1, name)
  from reversed names;
quit;
Output:
              name
          Fred Jones
```

Kate Kavich

Ron Turley

Yolanda Dulix

# **Search and Extract**

```
data null;
  length first last phone $ 16;
  retain re;
  if N = 1 then do;
   re = prxparse("/\(([2-9]\d\d)\) ?" ||
                 "[2-9]\d\d-\d\d\d'");
  end;
  input first last phone & $16.;
  if prxmatch (re, phone) then do;
    area code = prxposn(re, 1, phone);
    if area code ^in ("828" "336"
                       "704" "910"
                      "919" "252") then
    putlog "NOTE: Not in NC: "
           first last phone;
  end:
datalines;
Thomas Archer
                  (919)319-1677
Lucv Barr
                  (800)899-2164
Tom Joad
                  (508) 852-2146
Laurie Gil
                  (252)352 - 7583
Output:
   NOTE: Not in NC, Lucy Barr (800)899-2164
   NOTE: Not in NC, Tom Joad (508) 852-2146
```

For complete information refer to the Base SAS documentation at support.sas.com/base