

Ruby trunk - Bug #12728

Negative lookahead does not work for "+" even though works for "@"

09/06/2016 01:54 PM - rklemme (Robert Klemme)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	ruby 2.2.5p319 (2016-04-26 revision 54774) [x86_64-cygwin]	Backport: 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN

Description

I'll attach a test program that shows the effect. Basically, if I have a negative lookahead in the regex like (?!@) and "@" shows up in the proper location I get a mismatch (1. case). This is expected. If I exchange the "@" with a "+" or "[+]" in the regex and a "+" in the input, a match occurs (case 2 and 3). This is the bug. If the "+" or "@" is removed from the string an expected match occurs (case 4 and 5). I was not able to boil this down to a smaller example yet.

History

#1 - 09/30/2016 09:57 PM - naruse (Yui NARUSE)

- Status changed from Open to Rejected

In case 2, the regexp just behave as if

```
t %r{
  (?<!\)\)\( # outer bracket
  o\+

  (?<!\)\) ([+*]|\\{d+,\\}) (?!\+ ) # inner repetition, non possessive

  .*
  (?<!\)\)\) # outer bracket
  (?<!\)\) (?:[+*]|\\{d+,\\}) # unbounded repetition, non possessive

}x, "f(o++)+"
```

Of course it matches.

Maybe use should [a-zA-Z0-9]* or something instead of .*.

#2 - 10/01/2016 09:04 AM - rklemme (Robert Klemme)

- File rx-mini.rb added

Yui NARUSE wrote:

In case 2, the regexp just behave as if

```
t %r{
  (?<!\)\)\( # outer bracket
  o\+

  (?<!\)\) ([+*]|\\{d+,\\}) (?!\+ ) # inner repetition, non possessive

  .*
  (?<!\)\)\) # outer bracket
  (?<!\)\) (?:[+*]|\\{d+,\\}) # unbounded repetition, non possessive

}x, "f(o++)+"
```

Of course it matches.

Argh! Stupid me. Yes, the negative lookahead will also match with the closing bracket.

Maybe use should [a-zA-Z0-9]* or something instead of .*.

I included a closing bracket in the negative lookahead, then it works:

```
$ diff -U3 x1 x2
--- x1 2016-10-01 10:55:50.595060831 +0200
+++ x2 2016-10-01 10:55:44.459048792 +0200
@@ -13,20 +13,19 @@
```

/x

```
-MATCH
+NO MATCH
```

```
s = 'f(o++)+'
```

```
rx = /
```

```
(?<!)(                # outer bracket
```

```
(.*)
```

- (?<!)([+*]{\d+,}) (?!+) # inner repetition, non possessive

- (?<!)([+*]{\d+,}) (?!+)) # inner repetition, non possessive

```
(.)
```

```
(?<!))                # outer bracket
```

```
(?<!)(?:[+*]{\d+,}) # unbounded repetition, non possessive
```

/x

```
-match = #
```

```
MATCH
```

```
s = 'f(o++)+'
```

The following .* blinded me for the fact that the closing bracket can match with) AND fulfill the lookahead like in

```
irb(main):001:0> /a(?=b)b/.match "abc"
=> #
```

I am sorry for the hassle.

Kind regards

robert

PS: Attaching the test version that produced output x2.

Files

rx-mini.rb	1.63 KB	09/06/2016	rklemme (Robert Klemme)
rx-mini.rb	1.65 KB	10/01/2016	rklemme (Robert Klemme)