
Subject: bug in stregex?

Posted by [Vapuser](#) on Fri, 19 Jul 2002 20:54:07 GMT

[View Forum Message](#) <> [Reply to Message](#)

I don't you, you tell me. Is this a bug?

```
IDL> tt=stregex('cdefaz','(.*)(a|b|c)z',/extract)
IDL> help,tt
TT          STRING    = 'cdefaz'
```

My understanding of regular expressions says it's a bug. The first '(.*)' should only match up to the 'a' and the second subexpression, since it is **unqualified**, should handle the 'az', i.e. TT should have two parts. tt[0] should be 'cdef' and tt[1] should be 'a.'

If my regex had been '(.*)((a|b|c)z)*', the result would be understandable; then the 'greediness' of the '(.*)' regular expression should have consumed the whole string because the 'zero or more' qualifier applied to the second would have been satisfied, i.e. zero matches. But in my example the engine should have failed when attempting this match and should have backtracked two characters to produce the output I suggest above.

Perl certainly does it this way:

```
% perl
$s="cdefaz";
print "s=$s\n";
@tt=($s =~/(.*)(a|b|c)z/);
foreach (@tt) { print "$_\n";}
```

this code produces the output:

```
s=cdefaz
cdef
a
```

i.e. \$tt[0] = cdef and \$tt[1] = a, as expected.

But regular expression are always somewhat mysterious. Am I missing something here?

Comments?

whd
--

