

Infinite Recursion

Regular expressions such as `(?R)?z` or `a?(?R)?z` or `a|(?R)z` that use recursion without having anything that must be matched in front of the recursion can result in infinite recursion. If the regex engine reaches the recursion without having advanced through the text then the next recursion will again reach the recursion without having advanced through the text. With the first regex this happens immediately at the start of the match attempt. With the other two this happens as soon as there are no further letters `a` to be matched.

JGsoft V2 and Boost 1.64 treat the first two regexes as a syntax error because they always lead to infinite recursion. They allow the third regex because that one can match `a`. Ruby 1.9 and later, all versions of PCRE, and PCRE2 10.20 and prior treat all three forms of potential infinite recursion as a syntax error. Perl, PCRE2 10.21 and later, and Boost 1.63 and prior allow all three forms.

Circular Infinite Subroutine Calls

Subroutine calls can also lead to infinite recursion. All flavors handle the potentially infinite recursion in `((?1)?z)` or `(a?(?1)?z)` or `(a|(?1)z)` in the same way as they handle potentially infinite recursion of the entire regex.

But subroutine calls that are not recursive by themselves may end up being recursive if the group they call has another subroutine call that calls a parent group of the first subroutine call. When subroutine calls are forced to go around in a circle that too leads to infinite recursion. Detecting such circular calls when compiling a regex is more complicated than checking for straight infinite recursion. Only JGsoft V2 and Ruby 1.9 and later are able to detect this and treat it as a syntax error. All other flavors allow these regexes.

Errors and Crashes

When infinite recursion does occur, whether it's straight recursion or subroutine calls going in circles, JGsoft V2, Perl, and PCRE2 treat it as a matching error that aborts the entire match attempt. Boost 1.64 handles this by not attempting the recursion and acting as if the recursion failed. If the recursion is optional then Boost 1.64 may find matches where other flavors throw errors.

Boost 1.63 and prior and PCRE 8.12 and prior crash when infinite recursion occurs. This also affects Delphi up to version XE6 and PHP up to version 5.4.8 as they are based on older PCRE versions.

Endless Recursion

A regex such as `a(?R)z` that has a recursion token that is not optional and is not have [an alternative](#) without the same recursion leads to endless recursion. Such a regular expression can never find a match. When `a` matches the regex engine attempts the recursion. If it can match another `a` then it has to attempt the recursion again. Eventually `a` will run out of letters to match. The recursion then fails. Because it's not optional the regex fails to match.

JGsoft V2 and Ruby detect this situation when compiling your regular expression. They flag endless recursion as a syntax error. Perl, PCRE, PCRE2, and Boost do not detect endless recursion. They simply go through the matching process which finds no matches.

Make a Donation

Did this website just save you a trip to the bookstore? Please [make a donation](#) to support this site, and you'll get a **lifetime of advertisement-free access** to this site!

Page URL: <https://www.regular-expressions.info/recurseinfinite.html>

Page last updated: 30 March 2020

Site last updated: 08 April 2020

Copyright © 2003-2020 Jan Goyvaerts. All rights reserved.