Ruby - Bug #17341

Unsound quantifier reduction with nested quantifiers

11/23/2020 06:03 PM - jirkamarsik (Jirka Marsik)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	ruby 2.7.2p137 (2020-10-01 revision 5445e04352) [x86_64-linux]	**Backport:**	2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN

Description

The rules for reducing nested quantifiers can produce quantifiers with semantics which differ from the original quantifiers. This can then lead to the regular expressions matching different strings.

```
irb(main):001:0> /(?:a+?)*/.match('aa')
(irb):1: warning: nested repeat operator '+?' and '*' was replaced with '+? and ?' in regular expr
ession: /(?:a+?)*/
=> #<MatchData "a">
irb(main):002:0> /(a+?)*/.match('aa')
=> #<MatchData "aa" 1:"a">
```

In the above, we can see that by inserting a capture group between the two quantifiers, we prevent quantifier reduction from occurring and we get a regexp that matches the whole input. If we let quantifier reduction happen, we get a resulting regexp that only matches the first character. I think quantifier reduction should not change the behavior of a regexp, as it is just an optimization.

I found the quantifier reduction rules in ReduceTypeTable in regparse.c. I haven't checked them all but the ones that replace two quantifiers by two other quantifiers caught my eye.

Associated revisions

Revision 9e73177d5362c1986814f411961b712967dc5f97 - 12/02/2020 05:42 PM - jeremyevans (Jeremy Evans)

Do not reduce quantifiers if it affects which text will be matched

Quantifier reduction when using +?)* and +?)+ should not be done
as it affects which text will be matched.

This removes the need for the RQ_PQ_Q ReduceType, so remove the
enum entry and related switch case.

Test that these are the only two patterns affected by testing all
quantifier reduction tuples for both the captured and uncaptured
cases and making sure the matched text is the same for both.

Fixes [Bug #17341]

Revision 9e73177d5362c1986814f411961b712967dc5f97 - 12/02/2020 05:42 PM - jeremyevans (Jeremy Evans)

Do not reduce quantifiers if it affects which text will be matched

Quantifier reduction when using +?)* and +?)+ should not be done
as it affects which text will be matched.

This removes the need for the RQ_PQ_Q ReduceType, so remove the
enum entry and related switch case.

Test that these are the only two patterns affected by testing all
quantifier reduction tuples for both the captured and uncaptured
cases and making sure the matched text is the same for both.

Fixes [Bug #17341]

Revision 9e73177d - 12/02/2020 05:42 PM - jeremyevans (Jeremy Evans)

Do not reduce quantifiers if it affects which text will be matched

Quantifier reduction when using +?)* and +?)+ should not be done as it affects which text will be matched.

This removes the need for the RQ_PQ_Q ReduceType, so remove the
enum entry and related switch case.

Test that these are the only two patterns affected by testing all
quantifier reduction tuples for both the captured and uncaptured
cases and making sure the matched text is the same for both.

Fixes [Bug #17341]

History

#1 - 11/23/2020 11:58 PM - jeremyevans0 (Jeremy Evans)

I agree that the presence or absence of a capture group should not affect the matched text. Thank you for your spot on analysis regarding the ReduceTypeTable. By adding a test, I was able to determine this affect the +?)+ reduction in addition to the +?)* reduction, but those seem to be the only reductions affected, at least by the test I wrote. Maybe a different test would show errors in additional cases. I have submitted a pull request with a fix: https://github.com/ruby/ruby/pull/3808

#2 - 11/24/2020 11:17 AM - jirkamarsik (Jirka Marsik)

Thanks for the quick reply! Your fix looks great.

#3 - 12/02/2020 05:42 PM - jeremyevans (Jeremy Evans)

- Status changed from Open to Closed

Applied in changeset git|9e73177d5362c1986814f411961b712967dc5f97.

Do not reduce quantifiers if it affects which text will be matched

Quantifier reduction when using +?)* and +?)+ should not be done
as it affects which text will be matched.

This removes the need for the RQ_PQ_Q ReduceType, so remove the
enum entry and related switch case.

Test that these are the only two patterns affected by testing all
quantifier reduction tuples for both the captured and uncaptured
cases and making sure the matched text is the same for both.

Fixes [Bug #17341]