Ruby trunk - Bug #11113

Time ranges cannot be used in case statements in 1.9+ but they could in 1.8.7

05/01/2015 03:15 PM - myronmarston (Myron Marston)

Status: Closed
Priority: Normal
Assignee: 
Target version: ruby -v: 1.9.2, 1.9.3, 2.0, 2.1, 2.2
Backport: 2.0.0: WONTFIX, 2.1: DONE, 2.2: DONE

Description
Given this script:

```ruby
rangle = (Time.now - 1000)...(Time.now + 1000)

case Time.now
when range then puts "in range"
else puts "not in range"
end
```

I'd expect it to print "in range". This worked on 1.8.7:

```
$ ruby -v time_range_problem.rb
ruby 1.8.7 (2013-12-22 patchlevel 375) [i686-darwin12.5.0]
in range
```

...but raises an exception in 1.9.2, 1.9.3, 2.0, 2.1 and 2.2:

```
time_range_problem.rb:4:in `each': can't iterate from Time (TypeError)
  from time_range_problem.rb:4:in `include?'
  from time_range_problem.rb:4:in `include?'
  from time_range_problem.rb:4:in `==='!
  from time_range_problem.rb:4:in `<main>''
```

It looks like Range#=== is broken. Given that === is a standard ruby protocol, I would not expect a built-in type to raise an error like this. IMO, === should either return true or false but should not raise an error. We rely upon === in RSpec to do matching and this behavior means that if users pass us a time range, things blow up. See https://github.com/rspec/rspec-mocks/issues/938 for an example.

Associated revisions

Revision 50421 - 05/03/2015 01:02 AM - nobu (Nobuyoshi Nakada)
range.c: covered for linear objects

* range.c (linear_object_p, range_include): test if covered for linear objects. [ruby-core:69052] [Bug #11113]

Revision 50421 - 05/03/2015 01:02 AM - nobu (Nobuyoshi Nakada)
range.c: covered for linear objects

* range.c (linear_object_p, range_include): test if covered for linear objects. [ruby-core:69052] [Bug #11113]

11/29/2017
range.c: covered for linear objects

- range.c (linear_object_p, range_include): test if covered for linear objects. [ruby-core:69052] [Bug #11113]

merge revision(s) 50421: [Backport #11113]

* range.c (linear_object_p, range_include): test if covered for linear objects. [ruby-core:69052] [Bug #11113]

merge revision(s) 50421: [Backport #11113]

* range.c (linear_object_p, range_include): test if covered for linear objects. [ruby-core:69052] [Bug #11113]

History

#1 - 05/01/2015 10:48 PM - avit (Andrew Vit)
The "can't iterate" error is simply because Time is float-based: there is no succ number.

Something like this could be a solution:

```ruby
module Range::RubyIssue11113
  def ==(other)
    case first
    when Time, Float then self.cover?(other)
    else super
    end
  end
end

class Range
  prepend RubyIssue11113
end
```

Is the difference between a..b and a...b even relevant for float ranges?
The "can't iterate" error is simply because Time is float-based: there is no succ number.

Floats don't have a succ method, and yet (2.3...3.2) === 3.0 works just fine. Actually, I'm surprised that succ is used for === at all...it implies that === is a O(N) operation over the size of the range. Seems like something like range.begin < other && range.end > other would perform better (constant time) and avoid this problem. I haven't looked into the implementation of cover? but is that basically what it does internally?

Good point, it looks like Float is already "covered" (I didn't check) so it's only Time that is the issue here.

Yes, that's exactly what cover? does, instead of iterating with include?.

range.c: covered for linear objects

- range.c (linear_object_p, range_include): test if covered for linear objects.  [Bug #11113]

Thanks for the prompt fix, nobu (Nobuyoshi Nakada)!!

On a side note, I wasn't notified by email of your fix even though I'm watching this issue so I get notified of updates. Do you know why that is?
ruby_2_1 r50581 merged revision(s) 50421.

#8 - 05/23/2015 06:52 PM - nagachika (Tomoyuki Chikanaga)
- Backport changed from 2.0.0: WONTFIX, 2.1: DONE, 2.2: REQUIRED to 2.0.0: WONTFIX, 2.1: DONE, 2.2: DONE

Backported into ruby_2_2 at r50630.