Ruby - Bug #11385

`==` with bidirectional/cyclic dependency

07/22/2015 02:54 AM - allenwq (Allen Wang)

Status:	Rejected		
Priority:	Normal		
Assignee:	core		
Target version:			
ruby -v:	2.2.1	**Backport:**	2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN

Description

```
class Something
  attr_accessor :friend
  def initialize(friend)
    self.friend = friend
  end

  def ==(other)
    friend == other.friend
  end
end


a = Something.new([])
b = Something.new([a])
a.friend = [b]

a == b
```

The above code returns true on OS X and Linux, by right it should give me a exception of stack level too deep.

(And on windows I can see the expected exception)

Related issues:

Related to Ruby - Bug #1448: [patch] Proper handling of recursive arrays	**Closed**	**05/09/2009**

History

#1 - 07/22/2015 05:54 AM - nobu (Nobuyoshi Nakada)

- Description updated

- Status changed from Open to Rejected

true is the expected result, and the same with x64-mswin64_120.

#2 - 07/22/2015 07:16 AM - lowjoel (Joel Low)

Hi Nakada-san:

```
D:\> ruby -v
ruby 2.2.3p147 (2015-07-04 revision 51143) [x64-mswin64_120]

D:\> ruby test.rb
test.rb:8:in `==': stack level too deep (SystemStackError)
        from test.rb:8:in `=='
        from test.rb:8:in `=='
        from test.rb:8:in `=='
        from test.rb:8:in `=='
        from test.rb:8:in `=='
        from test.rb:8:in `=='
        from test.rb:8:in `=='
        from test.rb:8:in `=='
         ... 2574 levels...
        from test.rb:8:in `=='
        from test.rb:8:in `=='
        from test.rb:8:in `=='
        from test.rb:17:in `<main>'
```

It seems to be a problem specific to Windows. The test suite passes (IIRC)

#3 - 07/22/2015 08:34 AM - Eregon (Benoit Daloze)

Should the result not be **false** instead?
These two instances each have a different @friend and it cannot be determined if they are equal so "false" seems a much safer answer.

Also, this seems inconsistent to how Comparable#== treat such cases:

```
class Something
  attr_accessor :friend
  def initialize(friend)
    self.friend = friend
  end
  include Comparable
  def <=>(other)
    friend <=> other
  end
end
a = Something.new([])
b = Something.new([a])
a.friend = [b]
p a == b # false (and not true like above!)
```

#4 - 07/22/2015 08:56 AM - Hanmac (Hans Mackowiak)

my version is "ruby 2.3.0dev (2015-07-21 trunk 51319) [x86_64-linux]"

and it does return true for me too

#5 - 07/24/2015 04:00 PM - Eregon (Benoit Daloze)

#1448 and the specs in
https://github.com/ruby/rubyspec/blob/master/core/array/shared/eql.rb
convinced me this is desirable behavior.

#6 - 07/24/2015 04:04 PM - Eregon (Benoit Daloze)

- Related to Bug #1448: [patch] Proper handling of recursive arrays added

Files

test.rb	226 Bytes	07/22/2015	allenwq (Allen Wang)