ruby -v:
1.9.3

RUBY_VERSION
=> "1.9.3"
begin
  stack = []
do_stuff = lambda do [&chain]
  puts "Expected: #{chain.object_id}" 
  chain.call
end

chain = lambda do
  if filter = stack.shift
    filter.call(&chain)
  end
end

puts "Expected: #{chain.object_id}" 
stack << do_stuff 
chain.instance_eval { @completed = false }
chain.call
end

echo $(RUBY_VERSION) | ruby -c "puts \"Expected: \#{chain.object_id}\"; stack = []; do_stuff = lambda do [&chain]; puts \"Expected: \#{chain.object_id}\"; chain.call"|ruby
true

Evaluating the instance_eval:

```
begin
  stack = []
do_stuff = lambda do [&chain]
  puts "Actual: #{chain.object_id}" 
  chain.call
end

chain = lambda do
  if filter = stack.shift
    filter.call(&chain)
  end
end

puts "Expected: #{chain.object_id}" 
stack << do_stuff 
chain.instance_eval { @completed = false }
chain.call
end
```

Expected: 2152379740 
Actual: 2152379520

History

#1 - 02/17/2012 04:26 AM - marcandre (Marc-Andre Lafortune)

Might be a good idea to keep things simple:
do_stuff = lambda do |&block|
    puts "Object id is #{block.object_id}"
end

chain = lambda{}

do_stuff.call(&chain)
chain.instance_eval{}
do_stuff.call(&chain)

Prints out:
Block's object id is 2152284220
Block's object id is 2152284140

Using an equivalent method to do_stuff doesn't exhibit the same problem. Here's a complete example:

def stuff(&block)
    puts "Object id is #{block.object_id}"
end
do_stuff = method(:stuff).to_proc

chain = lambda{}

stuff(&chain)
do_stuff.call(&chain)
chain.instance_eval{}
stuff(&chain)
do_stuff.call(&chain)

Object id is 2156158200
Object id is 2156158200
Object id is 2156158200
Object id is 2156157980

#2 - 03/11/2012 05:33 PM - ko1 (Koichi Sasada)
- Assignee set to nobu (Nobuyoshi Nakada)

#3 - 03/18/2012 06:46 PM - shyouhei (Shyouhei Urabe)
- Status changed from Open to Assigned

#4 - 12/07/2012 03:49 PM - nobu (Nobuyoshi Nakada)
=begin
I don't think ((|pr2|)) is not guaranteed to be the same object as ((|pr1|)) where
pr1 = proc {}
pr2 = proc {|&pr| pr}.call(&pr1)
=end

#5 - 12/30/2012 10:46 PM - nobu (Nobuyoshi Nakada)
- Category set to core
- Assignee changed from nobu (Nobuyoshi Nakada) to matz (Yukihiro Matsumoto)
- Target version set to 3.0

#6 - 11/21/2018 08:06 AM - mame (Yusuke Endoh)
- Status changed from Assigned to Closed

It looks already fixed. I'm unsure if this behavior is guaranteed, though. Whether it is guaranteed or not, I think we can close this issue.