Ruby - Bug #11188

Method#inspect for chaining alias methods

05/27/2015 06:39 PM - ko1 (Koichi Sasada)

Status:	Closed		
Priority:	Normal		
Assignee:	matz (Yukihiro Matsumoto)		
Target version:			
ruby -v:	2.3dev	**Backport:**	2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN

Description

The following script prints strange results (at least for me).

```
class C0
  def foo
  end
end

class C1 < C0
  alias foo1 foo
  alias foo2 foo1
  alias foo3 foo2
end

p C1.new.method(:foo)
p C1.new.method(:foo1)
p C1.new.method(:foo2)
p C1.new.method(:foo3)
```

Result:

```
#<Method: C1(C0)#foo>
#<Method: C1(C0)#foo1(foo)>
#<Method: C1#foo2(foo)>
#<Method: C1#foo3(foo)>
```

I believe 3rd and 4th results should be:

```
#<Method: C1(C0)#foo2(foo)>
#<Method: C1(C0)#foo3(foo)>
```

How about it?

History

#1 - 08/12/2019 10:23 PM - jeremyevans0 (Jeremy Evans)

- File method-inspect-chain-alias-11188.patch added

I agree with ko1's opinion. Attached is a patch that implements his proposal.

Note that this change cannot be made in method_inspect, because at that time, the necessary information has already been lost. To implement ko1's proposal, changes to rb_alias are required, so that the defined_class in the alias chain methods (foo2 and foo3) is set correctly to C0.

#2 - 05/24/2020 03:39 AM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Closed

This patch was merged in adecd43197d5ea2a62a618a5c9be653bcf009c62.

#3 - 05/24/2020 10:34 AM - Eregon (Benoit Daloze)

Nice, in fact this was already the behavior in TruffleRuby and JRuby.

Files

method-inspect-chain-alias-11188.patch	2.37 KB	08/12/2019	jeremyevans0 (Jeremy Evans)