Refinements cannot refine method_missing and respond_to_missing?

01/14/2017 05:21 PM - matsuda (Akira Matsuda)

Class C

```ruby
using Module

def refine C do
  def x() p:x; end
  def method_missing(m, *args)
    m == :foo ? p!(:fooo!) : super
  end
  def respond_to_missing?(m, include_private = false)
    m == :foo || super
  end
end

C.new.x
p C.new.respond_to? :foo
C.new.foo
```

The script above doesn’t respond_to :foo nor run :foo as expected. Actually, the result differs between ruby versions.

```ruby
% ruby -v t.rb
ruby 2.5.0dev (2017-01-14 trunk 57328) [x86_64-darwin15]
:x
false
t.rb:19:in ':<main>': undefined method `foo' for #<C:0x007f90ca0fb240> (NoMethodError)
% ruby -v t.rb
ruby 2.4.0p0 (2016-12-24 revision 57164) [x86_64-darwin15]
:x
false
t.rb:19:in ':<main>': undefined method `foo' for #<C:0x007f80ae097780> (NoMethodError)
% ruby -v t.rb
ruby 2.3.3p222 (2016-11-21 revision 56859) [x86_64-darwin15]
:x
false
t.rb:19:in ':<main>': undefined method `foo' for #<C:0x007fd89c83b518> (NoMethodError)
% ruby -v t.rb
ruby 2.2.6p396 (2016-11-15 revision 56800) [x86_64-darwin15]
:x
false
:fooo!
% ruby -v t.rb
ruby 2.1.10p492 (2016-04-01 revision 54464) [x86_64-darwin15.0]
:x
false
:fooo!
```
What I can tell is that method_missing was broken at somewhere in between 2.2 and 2.3, and respond_to_missing? has never worked correctly.

Akira Matsuda wrote:

Refinements with method_missing and respond_to_missing? behaves very strangely.

```ruby
class C;
end
using Module.new {
  refine C do
    def x() p:x; end
    def method_missing(m, *args)
      m == :foo ? p(:fooo!) : super
    end
    def respond_to_missing?(m, include_private = false)
      m == :foo || super
    end
  end
};
C.new.x
p C.new.respond_to? :foo
C.new.foo
```

The script above doesn't respond_to :foo nor run :foo as expected. Actually, the result differs between ruby versions.

I think the behavior of 2.3 or later is expected because *_missing are indirect method calls.

Matz, what do you think of changing the behavior?

#4 - 12/19/2017 06:37 AM - naruse (Yui NARUSE)
- Target version deleted (2.5)

#5 - 12/20/2017 12:00 AM - matz (Yukihiro Matsumoto)
- Backport set to 2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN

https://github.com/nobu/ruby/tree/feature/13129-refine-method_missing
Currently, I feel negative about the proposal.

First, as Shugo commented, method_missing etc. are considered indirect method calls. Second, mixing refinements (static rebinding) and method-missing (dynamic rebinding) are very complex and easily become complex.

This is not the final decision. You may be able to persuade me in the future. But this is my current opinion.

Matz.

#6 - 12/20/2017 12:01 AM - matz (Yukihiro Matsumoto)
- Backport deleted (2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN)
- ruby -v deleted (ruby 2.5.0dev (2017-01-14 trunk 57328) [x86_64-darwin15])
- Target version deleted (2.5)
- Tracker changed from Bug to Feature